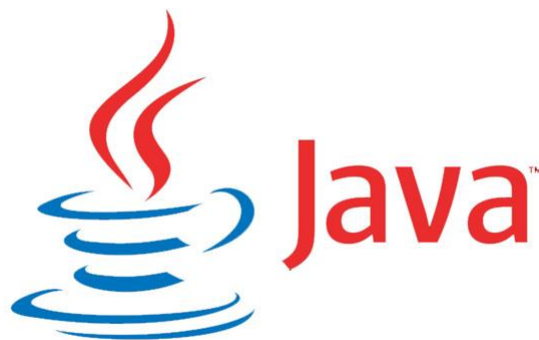


Schule Informationstechnik  
der Bundeswehr



**Sprachausbildung Java**



**Übungen**  
**"Grundlagen der Programmierung"**

# Grundlagen

## 1 Grundbegriffe

Erläutern Sie kurz die folgenden Begriffe:

- a) Betriebssystem
- b) Algorithmus
- c) Befehl
- d) Programm
- e) Programmiersprache
- f) Maschinensprache
- g) Assemblersprache
- h) Höhere Programmiersprachen
- i) Programmierung
- j) Quellcode
- k) Sourcecode
- l) Compiler
- m) Interpreter
- n) Virtuelle Maschine

## 2 Compiler

Was ist der Unterschied zwischen den Produkten eines Java-Compilers und eines C++-Compilers?

## 3 Literale

Von welchem Typ sind jeweils die folgenden Literale? Eines dieser Literale ist übrigens falsch, welches? (Der Logiker würde vielleicht sogar zwei finden.)

a) 45	i) 1386
b) 5.3	j) 0x45L
c) 9.2e5f	k) \"
d) 7L	l) 0x4305F
e) True	m) 0x4305FF
f) 'a'	n) 0371
g) 083	o) 0x4305FFL
h) 5.0F	p) false

## 4 Logik

a und b seien vom Typ Boolean (bool). Zeigen Sie, dass folgender Ausdruck immer wahr ist:

$$(!a \parallel !b) == !(a \&\& b)$$

Der Beweis lässt sich am besten mit einer Wahrheitstabelle führen:

a	b	!a	!b	!a  !b	a&& b	!(a&&b)
true	true	false	false			
true	false	false	true			
false	true					
false	false					

## 5 Zahlenformate (binär, dezimal, hexadezimal, negativ)

a) Geben Sie die Binärdarstellung für die folgenden Dezimalzahlen an:

- 7:
- 20:
- 51:
- 121:
- 514:
- 1023:

b) Geben Sie die Dezimaldarstellung folgender Binärzahlen

- 0010 1110:
- 1010 0101:
- 1001 1010:
- 1100 1101:

c) sowie dieser Hexadezimalzahlen an.

- 39:
- 1A:
- CA:
- FE:

d) Geben Sie die Binärdarstellung (8 Bit) für die folgenden Dezimalzahlen an:

- 1:
- 21
- 51:
- 128:

## 6 Zahlenformate (binär, ternär, dezimal)

Gegeben sei die Zahl 1001 1001. Welchen dezimalen Wert hat Sie, wenn man sie

- als positive ganze Zahl in binär-Darstellung
- als ganze Zahl in binär-Darstellung mit Vorzeichenbit
- als positive Ternärzahl (d.h. im Dreiersystem)

interpretiert?

## Einfache Programme

### 7 Ausgabe in der Konsole

Schreiben Sie ein Programm, bei dem folgende Ausgabe auf dem Bildschirm erscheinen soll.

```
Hallo Herr <Ihr Name>!  
Willkommen zum Lehrgang Anwendungsprogrammierer der Bundeswehr Java
```

Kompilieren und führen Sie das Programm aus.

### 8 Ausgabe in der Konsole

Schreiben Sie ein Programm, welches die Lösung der Aufgabe 5 auf dem Monitor ausgibt.

Kompilieren Sie das Programm und führen es aus.

### 9 Logische Operationen mit Ganzzahlen

Welchen Wert haben die Ausdrücke in den Zeilen 3-6?

```
1  int a = 21;  
2  int b = 28;  
3  a & b;  
4  a ^ b;  
5  a >> 2;  
6  a && b;
```

Überlegen Sie zuerst selbst. Lassen Sie sich anschließend die Lösung von einem Programm in der Konsole ausgeben.

### 10 Zeit umrechnen

Ein Satellit funkt Zeitspannen als „Anzahl Sekunden“ zur Erde. Schreiben Sie ein Programm `SatelliteTime`, das einen Sekundenbetrag einliest und die Zeitspanne in der Form „d h m s“ wieder ausgibt, wobei gilt:

- d = Anzahl Tage,
- h = Anzahl Stunden im Bereich 0 bis 23,
- m = Anzahl Minuten im Bereich 0 bis 59,
- s = Anzahl Sekunden im Bereich 0 bis 59.

Teilen Sie das Programm deutlich in die 3 Bereiche Eingabe, Verarbeitung / Berechnung und Ausgabe auf.

Test Sie das Programm beispielweise mit dem Aufruf:

```
Geben Sie eine Sekundenanzahl ein:  
10000  
0 2 46 40
```

## 11 Summe und Durchschnitt

Erstellen Sie ein Java-Programm, das drei Gleitpunktzahlen im Dialog einliest und deren Summe und Durchschnitt ausgibt.

*Beispielausgabe:*

```
Geben Sie drei Gleitpunktzahlen ein:  
2.7  
8.9  
5.3  
Die Summe:           16,9  
Der Durchschnitt:    5,63333
```

Teilen Sie das Programm in Eingabe, Verarbeitung und Ausgabe auf. Es sollen keine Berechnungen in der Ausgabe stattfinden.

## 12 Temperatursysteme (Formatierte Ausgabe)

Schreiben Sie ein Java-Programm, das

- a) eine Fahrenheit-Temperatur im Dialog einliest und in Celsius umrechnet,
- b) eine Celsius-Temperatur einliest und in Fahrenheit umrechnet

und die Ergebnisse mit zwei Nachkommastellen ausgibt. Denken Sie daran die Bereiche Eingabe, Verarbeitung und Ausgabe voneinander zu trennen. Formel:  $^{\circ}\text{C} = (^{\circ}\text{F} - 32) * 5/9$

*Beispielausgabe:*

```
Geben Sie eine Temperatur in Fahrenheit ein:  
100  
100,00 Fahrenheit entsprechen 37,78 Grad Celsius.  
  
Geben Sie eine Temperatur in Celsius ein:  
28  
28,00 Celsius entsprechen 82,40 Grad Fahrenheit.
```

## 13 Wertetausch

```
1 public static void main(String[] args) {  
2     int a=3;  
3     int b=4;  
4     System.out.println(a+" "+b);  
5     /*  
6         Vertauschen Sie hier a und b  
7     */  
8     System.out.println(a+" "+b);  
9 }
```

Vertauschen Sie in diesem Programm die Werte der Variablen a und b. Der Rahmen (Zeilen 1 bis 4 und 8 bis 9) wird hierbei nicht verändert.

- a) Variante: Nutzen einer weiteren Variablen.
- b) Variante: Ohne zusätzliche Variable.

Welche Variante ist verständlicher? Welche Variante ist besser lesbar?

## 14 Bananas

Schreiben Sie ein Programm `Bananas`, das als erste Zahl den Brutto-Kilopreis von Bananen in Cent einliest. Als zweite Zahl liest das Programm das Gewicht einer Menge von Bananen in Gramm. Beide sind ganze, nicht-negative Zahlen. Daraufhin gibt das Programm den Bruttopreis aus, wie im untenstehenden Beispiel gezeigt.

Erweitern Sie Ihr Programm so, dass es auch den Nettopreis und die enthaltene Mehrwertsteuer ausgibt. Der Mehrwertsteuersatz in Prozent wird als dritte Zahl eingelesen.

`Bananas` liefert beispielweise folgende Ergebnisse:

```
Geben Sie den Kilopreis in Cent ein:
232
Geben Sie das Gewicht in Gramm ein:
1500
Geben Sie die MWSt in Prozent ein:
16
Brutto      3,48 Euro
MWSt        0,48 Euro
Netto       3,00 Euro
```

Achten Sie darauf, dass

1. sich aus Nettopreis und Steuersatz der enthaltene Steuerbetrag ergibt und dass
2. die Summe aus Nettopreis und Steuerbetrag exakt den Bruttopreis ergibt.

Die Ausgabe wird auf ganze Cent gerundet. Ab 0,5 wird auf- und darunter abgerundet.

Trennen Sie Eingabe, Berechnung und Ausgabe.

## Kontrollstrukturen

### 15 Rechnen am Kreis

Schreiben Sie ein Programm, welches den Benutzer danach fragt, ob der Umfang oder der Flächeninhalt eines Kreises berechnet werden soll. Anschließend soll das Programm den Radius einlesen und das Ergebnis der gewählten Berechnung ausgeben. Beispiele:

```
Wollen Sie den Umfang (u) oder den Flächeninhalt (A) berechnen:  
u  
Radius:  
2  
Umfang:  $u = 2 * \text{PI} * r = 2 * \text{PI} * 2 = 12,57$ 
```

```
Wollen Sie den Umfang (u) oder den Flächeninhalt (A) berechnen:  
A  
Radius:  
2  
Flächeninhalt:  $A = \text{PI} * r^2 = \text{PI} * 2^2 = 12,57$ 
```

Denken Sie an die Trennung von Eingabe, Verarbeitung und Ausgabe.

### 16 Vokale erkennen

Schreiben Sie ein Programm, welches den Nutzer nach einem Buchstaben fragt und ausgibt, ob es sich bei dem Buchstaben um einen Vokal handelt oder nicht. Beispiel:

```
Buchstabe:  
a  
Eingabe war ein Vokal: A
```

```
Buchstabe:  
b  
Eingabe war kein Vokal.
```

Implementieren Sie das Programm einmal mit einem If-Konstrukt und einmal als Switch-Case-Konstrukt.

### 17 Median ermitteln

Schreiben Sie ein Java-Programm, das drei Ganzzahlen im Dialog einliest, ihren Median berechnet und ausgibt. Der Median von drei Ganzzahlen ist der mittlere Wert dieser Zahlen.

*Hinweis:* Der Median von 7, 3 und 9 z.B. ist 7. (Wem das Beispiel noch nicht hilft, kann ja Google mal fragen.)

## 18 Intervallprüfung

Es soll eine Zahl eingegeben und geprüft werden, ob diese Zahl in einem festgelegten Intervall ( $5 < x < 100$ ) ist. Wenn dies der Fall ist, wird die Zahl ausgegeben, ansonsten *"falsche Eingabe"*. Denken Sie an die Trennung von Eingabe, Verarbeitung und Ausgabe.

## 19 Rabatt

Ein Einkaufsladen möchte ein neues Rabattsystem implementieren. Ab einem Einkaufswert von 100 € bekommt der Kunde 2% Rabatt, ab einem Einkaufswert von 300 € bekommt der Kunde 3 % Rabatt und ab einem Einkaufswert von 1000 € erhält der Kunde 50 € Rabatt.

Schreiben Sie ein Programm, welches nach dem Einkaufswert fragt, die entsprechende Rabattgruppe, den Rabatt und den zu zahlenden Betrag anzeigt. Beispiel:

```
Einkaufswert:
333
Rabattgruppe: 3 %
Rabatt:      9,99 €
Betrag:      323,01 €
```

Denken Sie an die Trennung von Eingabe, Verarbeitung und Ausgabe.

## 20 Zahlenraten

Schreiben Sie ein Programm, welches den Nutzer auffordert eine Zahl zu erraten. Der Nutzer darf maximal 3-mal raten. Hat er die richtige Zahl erraten, bekommt er dies mitgeteilt und das Programm wird beendet. Beispiel:

```
Bitte raten Sie:
1
Bitte raten Sie erneut:
2
Bitte raten Sie ein letztes Mal:
3
Leider nicht gewonnen
```

```
Bitte raten Sie:
5
Sie haben richtig geraten
```



## 21 BMI-Rechner

Der Body-Mass-Index (BMI) ist eine Maßzahl zur Bewertung des Körpergewichts eines Menschen in Relation zu seiner Körpergröße. Der BMI bezieht die Körper-Masse (kg) auf das Quadrat der Körpergröße (m<sup>2</sup>).

$$\text{Formel: } BMI = \frac{m}{h^2} = \frac{[kg]}{[m^2]}$$

Interpretation:

Kategorie	BMI [kg/m <sup>2</sup> ]	Körpergewicht
starkes Untergewicht	<16	Untergewicht
mäßiges Untergewicht	16 - <17	
leichtes Untergewicht	17 - <18,5	
Normalgewicht	18,5 - <25	Normalgewicht
Präadipositas	25 - <30	Übergewicht
Adipositas Grad I	30 - <35	Adipositas
Adipositas Grad II	35 - <40	
Adipositas Grad III	ab 40	

Schreiben Sie ein Programm, welches die Größe und die Masse eines Menschen abfragt und die Kategorie sowie den errechneten BMI mit 2 Nachkommastellen ausgibt. Beispiel:

```
Körpergröße (m) :
1.8
Körpermasse (kg) :
80
BMI = 24,69 (kg/m^2)
Kategorie = Normalgewicht
```

## 22 Notenberechnung

Schreiben Sie ein Programm, welches Schulnoten nach den folgenden Bedingungen berechnet und ausgibt:

$$100 \text{ Prozent} = \text{Note: } 1,0$$

$$100 > x \geq 50 \text{ Prozent} = \text{Note: } 10,99 - 0,1 * x$$

$$x < 50 \text{ Prozent} = \text{Note: } 6,0$$

Nutzen Sie if- und Switch-Case-Konstrukte und denken Sie an die Trennung zwischen Eingabe, Verarbeitung und Ausgabe.

## 23 Punkte zu Geld machen

Drei Spieler haben jeweils einen bestimmten Punktestand erreicht. (Um welches Spiel es sich handelt, spielt hier gar keine Rolle.) Lesen Sie diese Werte ein.

Der Spieler, welcher mehr Punkte hat als ein anderer Spieler, bekommt von ihm die Differenz der Punkte in Euro ausbezahlt. Geben Sie aus, wer wem wieviel bezahlen muss.

*Beispiel:*      SpielerA: 50 Punkte  
                    SpielerB: 100 Punkte  
                    SpielerC: 20 Punkte

*Lösung:*        SpielerA zahlt 50€ an SpielerB  
                    SpielerC zahlt 30€ an SpielerA  
                    SpielerC zahlt 80€ an SpielerB

## 24 Analyse mit Schafen

### Brüderlich Teilen

"Zwei Brüder erbten eine Herde Schafe. Sie verkauften alle Tiere und bekamen für jedes Schaf genauso viele Euro, wie Schafe in der Herde waren."

„Sie erhielten das Geld in 10 € - Scheinen, der Restbetrag wurde in 1 € - Münzen ausgezahlt. Sie teilten das Geld unter sich auf, indem sie es auf einen Tisch legten und immer abwechselnd einen Schein nahmen, bis keiner mehr übrig war. Der ältere Bruder durfte anfangen."

### Aufgabe

Schreiben Sie ein Programm, das die Zahl der Schafe einliest und berechnet, wie viel Geld die beiden Brüder gemäß dem obigen Verfahren jeweils bekommen haben.

### Zusatz

Da der ältere Bruder anfangen durfte, bekam er auch immer den letzten Schein, also in manchen Fällen 10€ mehr. Um das Ungleichgewicht wenigstens teilweise auszugleichen gab er dem jüngeren die Euromünzen. Der war jedoch nicht zufrieden, da er damit ja weniger als 10€ bekommen hat.

Der ältere sah dies ein: "Ich werde dir einen Scheck über einen passenden Betrag ausschreiben, so dass wir am Schluss beide genau die gleiche Summe besitzen."

**Auf welche Summe lautete der Scheck?**

## Schleifen

### 25 Schleifenköpfe

Formulieren Sie die für die folgenden Aufgabenstellungen angemessenen for – Schleifen. Lassen Sie dabei den "Schleifenkörper" leer, es genügt der Schleifenkopf, d.h.

1     **for (Initialisierung; Bedingung; Veränderung) {...}**

- Addieren sie die Zahlen von 1 – 100.
- Addieren Sie die Quadrate aller Zahlen kleiner 20.
- Summieren Sie die Kehrwerte aller geraden Zahlen kleiner 50.
- Lesen Sie eine ganze Zahl n ein und addieren Sie die Wurzeln der Zahlen von 1 bis n.
- Multiplizieren Sie eine Folge von ganzen Zahlen. Lesen Sie Anfangs- und Endwert ein.
- Lesen Sie n ein und geben Sie die ersten n Primzahlen aus.

### 26 Zahlenpyramide

Lesen Sie eine Zahl n ein und geben Sie jede Zahl i von 1 bis n (d.h.  $1 \leq i \leq n$ ) genau i mal aus. Nutzen Sie für jede Zahl eine Zeile.

*Ergebnis:*

```
Zahl:
n

1
2 2
3 3 3
...
n n ... n
```

Lösen Sie die Aufgabe mit:

- while Schleifen
- for Schleifen
- do-while Schleifen

*Zusatzaufgabe für Vollprofis:*

Sorgen Sie durch geeignet positionierte Leerzeichen dafür, dass sich eine Pyramide ergibt:

```
    1
  2  2
3 3 3
...
```

**Hinweis:** Beginnen Sie zunächst mit einstelligem n.

## 27 Wiederholte Notenberechnung

Erweitern Sie die Lösung von Aufgabe 22 so, dass automatisch nach jeder Notenberechnung gefragt wird, ob eine weitere Note berechnet werden soll. Nutzen Sie dazu eine do-while-Schleife.

## 28 Quadratsummen

Schreiben Sie ein Programm, das die Summe der Quadrate der  $n$  positiven Ganzzahlen berechnet und ausgibt.

Formel:

$$\text{Summe} = 1^2 + 2^2 + 3^2 + \dots + n^2$$

Zusatz:

Das Programm soll sich nach dem berechnen nicht beenden, sondern nach der nächsten Zahl  $n$  fragen.

## 29 Prüfung auf Primzahl

Schreiben Sie ein Programm, das eine positive ganze Zahl einliest und prüft, ob sie eine Primzahl ist. Eine sinnvolle Ausgabe wird erwartet.

Eine Primzahl ist eine natürliche Zahl, die

1. größer als 1 ist.
2. nur durch sich selbst und 1 ohne Rest teilbar ist.

**Beispiele:** 2, 3, 5, 7, 11, 13, 17, 19, 23...

**Hinweis:** Nicht direkt loshacken, sondern erst mal über den Algorithmus nachdenken

## 30 Fakultät berechnen

Sie kennen sicherlich aus der Schulmathematik die Fakultätsfunktion.

Zur Erinnerung:

$$n! = n * (n-1) * (n-2) * \dots * 1$$

Als Beispiel:

$$n = 5$$

$$5! = 5 * 4 * 3 * 2 * 1 = 120$$

Erstellen Sie ein Programm `Fakultaet`, das  $n$  einliest und  $n!$  berechnet und ausgibt.

## Reihungen

### 31 Arrays erstellen

Definieren Sie ein Array, das

- das Monatsgehalt von 20 Angestellten speichern kann. Die ersten beiden Elemente werden mit 3000.00 initialisiert.
- fünf Ganzzahlen speichern kann. Als Anfangswert erhält jedes Element das Doppelte seines Indexwertes.
- eine Liste von Namen speichern kann. Die Namen sind: Jody Foster, Bon Jovi, Frank Zappa.

### 32 Messwert-Reihung

In einer Reihung von 20 Feldern sind Messwerte eines Versuches gespeichert. Die Messwerte sind ganzzahlig und liegen zwischen 0 und 1000. Zu befüllen der Komponenten bedienen Sie sich dem Zufallsgenerator. Bestimmen Sie das Minimum und Maximum der Messwerte im Datentyp der Messwerte, sowie den Durchschnittswert aller Messwerte als double-Zahl.

Treffen Sie die notwendigen Vereinbarungen.

Ergänzung:

Stellen Sie den Minimum- und Maximum-Wert graphisch mit Sterne ,\*' dar. Dabei soll die Skalierung in y-Achse nach oben vom Werte-Typ: 1 Zeile (Stern) pro 100 darstellen. Die beiden Werte sollen zuerst untereinander und danach nebeneinander erscheinen. In x-Achse soll für jeden Wert eine Balkenstärke von 3 Sternen gewählt werden und Abstand zwischen Balken soll 2 Leerzeichen betragen.

```
Min-Wert:      2
Max-Wert:     888
Durchschnittswert: 280,20
```

**Min-Wert**

```
1000
 900
 800
 700
 600
 500
 400
 300
 200
 100
  0  ***
```

**Max-Wert**

```
1000
 900
 800  ***
```

700	***	
600	***	
500	***	
400	***	
300	***	
200	***	
100	***	
0	***	
Min-/Max-Wert		
1000		
900		
800	***	
700	***	
600	***	
500	***	
400	***	
300	***	
200	***	
100	***	
0	***	***

### 33 Fibonacci Array

Die Folge der Fibonacci-Zahlen ist wie folgt definiert:

- die erste und zweite Fibonacci-Zahl sind gleich 1
- die folgenden Fibonacci-Zahlen ergeben sich aus der Summe der jeweils vorausgehenden zwei Werte, d.h.

$$F_1 \equiv 1$$

$$F_2 \equiv 1$$

$$F_j \equiv F_{j-1} + F_{j-2} \quad \text{für } j \geq 3$$

- Definieren Sie ein Reihungsobjekt zur Speicherung der Fibonacci-Zahlen bis **n-Zahlen**. Die Anzahl **n** soll vorab vom Nutzer abgefragt werden.
- Schreiben Sie ein Java-Programm, wodurch o.a. Reihungsobjekt mit den entsprechenden Fibonacci-Zahlen gefüllt wird und geben dieses anschließend aus.

## 34 Reiseunternehmen

Ein Reiseunternehmen organisiert zu einem Erholungsgebiet Busfahrten. Die Busfahrt findet an jedem Wochentag einmal statt und bietet 40 Personen Platz. Die Kunden haben die Möglichkeit Plätze für eine Woche im Voraus zu buchen.

Schreiben Sie ein Java-Programm, das diese Platzverwaltung realisiert. Das Programm bietet 4 Operationen an:

- **Buchen:** Belegung von  $x$  Plätzen an einem Wochentag. Kann eine Belegung nicht komplett durchgeführt werden, ist eine entsprechende Meldung auszugeben und nicht zu buchen.
- **Storno:** Freigabe von  $x$  Plätzen an einem Wochentag. Sind weniger als  $x$  Plätze im Bus überhaupt belegt, ist eine entsprechende Meldung auszugeben. Eine Teilfreigabe ist nicht durchzuführen.
- **Neu:** Neuinitialisierung eines Busses an einem Wochentag mit 40 freien Plätzen.
- **Ende:** Beendet das Programm

Eine Überprüfung der Eingabe  $x$  ist nicht notwendig ( $1 \leq x \leq 40$ ).

Bieten Sie ein Menü an, das die vier Optionen darstellt, in dem der Nutzer eine Auswahl treffen kann.

### Hinweis:

Es wird ein Array-Objekt `bus` benötigt, das die Wochentage als Index und die Anzahl Plätze als Komponente besitzt.

## 35 Abzählreim

Bei manchen Kinderspielen scheidet mit Hilfe eines Abzählreimes (z.B. "Ene-mene-mu-und-raus-bist-Du") von  $m$  Mitspielern immer der  $n$ -te aus.

Stellvertretend für die Mitspieler stehen in einem Feld der Länge  $m$  Wahrheitswerte, die anzeigen, ob der entsprechende Mitspieler noch an der Abzählung teilnimmt oder bereits ausgeschieden ist.

Schreiben Sie ein Programm, das bei vorgegebenen Werten  $m$  und  $n$  einen Abzählreim – mit gleichzeitiger Ausgabe der ausscheidenden Mitspieler – simuliert.

## 36 Binäre Suche

Eine wesentliche Aufgabe in der Datenverarbeitung ist neben dem Sortieren auch das Suchen von Datensätzen in sortierten Datenbeständen.

Bei der linearen Suche durchsuchen wir eine Reihung vom Beginn und beenden die Suche, wenn der Datensatz gefunden ist oder nicht mehr enthalten sein kann.

Wesentlich besser eignet sich dagegen die `binäre Suche`, die auf die Mitte des Feldes zugreift. Anhand des dort gespeicherten Datensatzes kann entschieden werden, ob der

gesuchte Satz gefunden wurde oder in der ersten oder zweiten Hälfte der Reihung weitergesucht werden soll. Das Verfahren wird anschließend auf die ausgewählte Hälfte angewendet.

Schreiben Sie ein Programm, das einen Datensatz mittels binärer Suche sucht. Geben Sie sich zum Test ein gefülltes Reihungsobjekt vor, doch konzentrieren Sie sich darauf, den Algorithmus von vornherein allgemeingültig zu formulieren.

Beispiel: gesuchte Zahl  $\Rightarrow$  17

$$17 < 56 ?$$

1	2	3	4	5	6	7	8	9	10
8	17	21	34	56	59	65	100	103	236

$$17 < 17 ?$$

1	2	3	4
8	17	21	34

## 37 Hex nach Dez

Schreiben Sie ein vollständiges Java-Programm, das Hexadezimalzahlen in einen Zeichenpuffer einliest und den entsprechenden dezimalen Wert ausgibt.

Beispiel eines Dialog (Benutzereingaben in *kursiv*):

```
Geben Sie eine Hexadezimalzahl in der Form:
X'ABC'
ein:
X'ABC'
Ergebnis = 2748
```

Die Hexadezimalzahlen liegen in folgender Form (EBNF) vor:

$$\text{Hexadezimalzahl} ::= X' \text{Hexziffer} \{ \text{Hexziffer} \}'$$

$$\text{Hexziffer} ::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9 \mid A \mid B \mid C \mid D \mid E \mid F$$

Die eingegebenen hexadezimalen Zahlen sind auf formale Gültigkeit zu prüfen. Eine eventuelle Wertebereichsüberschreitung des dezimalen Äquivalents **Integer.MAX\_VALUE** kann vernachlässigt werden.

Hinweis:

Zur Umwandlung ist das Horner-Schema zu verwenden. D.h. Gültigkeitsprüfung und Rechenoperationen erfolgen zeichenweise in Notationsreihenfolge

Bsp. :  $ABC_{(16)} = (10 * 16 + 11) * 16 + 12$



### 38 Zweidimensionales Array (Reihungen über Reihungen)

Gegeben sind folgende Definitionen:

- Erstellen Sie ein eindimensionales Array `v1`, welches drei int-Werte verwalten kann und weisen Sie jeder Komponente eine 1 zu.
- Erstellen Sie, dass u.a. zweidimensionale Array `v2_1` und befüllen dies gleich mit den vorgegebenen Werten. Geben Sie die Anweisungsfolge zur Ausgabe aller skalaren Werte von `v2_1` an.
- Füllen Sie alle Komponenten von `v2_1` mit `v1`.
- Schreiben Sie eine Routine, die `v2_1` nach `v2_2` speichert; die Speicherung soll komponentenweise spiegelbildlich und innerhalb der Komponenten ebenfalls spiegelbildlich erfolgen.

Beispiel:

`v2_1` :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----

`v2_2` wird zu :

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
----	----	----	----	----	----	---	---	---	---	---	---	---	---	---

### 39 Quadratische Matrix

Beispiel für  $n=5$ :

Diagonale 1	5	2	9	32	56
	6	12	10	22	3
	23	5	21	6	17
	3	11	64	89	25
Diagonale 2	7	57	2	1	4

Erstellen Sie die dargestellte Matrix mit den vorgegebenen Werten und schreiben Sie eine Anweisungsfolge, die feststellt, welche Diagonale die niedrigere Summe hat.

## Statische Methoden

### 40 Methodenkopf

Geben Sie die **Methodenköpfe** folgender statischer Methoden an:

- Die Methode `sum()` liefert die Summe von drei double-Werten, die als Argumente übergeben werden.
- Die Methode `cubes()` besitzt einen Parameter `n` vom Typ `int`. Sie summiert die ersten `n` positiven Zahlen "hoch drei" auf, berechnet also den Wert  $1^3 + 2^3 + \dots + n^3$ , und gibt dieses Ergebnis auf der Konsole aus.
- Die Methode `ggt()` bestimmt den größten gemeinsamen Teiler von zwei als Parameter übergebenen ganzen Zahlen.
- Die Methode `isLeapYear()` erhält eine Jahreszahl als Argument und gibt `true` zurück, falls das Jahr ein Schaltjahr ist, andernfalls `false`.
- Die Methode `displayStatus()` gibt den Status des Programms auf dem Bildschirm aus. Es hat keinen Parameter und keinen Return-Wert.

### 41 Double Array

Erstellen Sie ein Programm, das im Dialog bis zu 10 Gleitpunktzahlen in einen Array mit Elementen vom Typ `double` einliest.

- Schreiben Sie anschließend eine statische Methode, die den größten oder kleinsten Wert berechnet und auf der Konsole ausgibt.
- Schreiben Sie eine Methode, die den Durchschnitt aller Elemente berechnet und das Ergebnis auf der Konsole ausgibt.

### 42 Zylindervolumen

Definieren Sie eine statische Methode `cylinderVolume()`, die das Volumen eines Zylinders berechnet. Die Methode erhält den Radius eines Kreises und die Höhe des Zylinders als Parameter vom Typ `double`.

*Hinweis:* Die Formel zur Berechnung des Zylindervolumens lautet  $V = \pi * \text{radius}^2 * \text{height}$  ( $\pi = 3.1415927$ )

Zum Testen der Methode schreiben Sie eine `main`-Methode, die das Volumen für die Radien und Höhen 0.5, 1.0, 1.5, 2.0 wie folgt ausgibt:

**** Volumen eines Zylinders ****		
Radius	Hoehe	Volumen
-----		
0,5	0,5	0,392699
0,5	1,0	0,785398
0,5	1,5	.
.	.	.

.	.	.
2,0	1,5	18,849556
2,0	2,0	25,132742

### 43 Quersumme

Schreiben Sie eine statische Methode, welche die Quersumme einer ganzen Zahl berechnet und diese zurückliefert.

### 44 Dezimal zu Binär

Wir haben am ersten Lehrgangstag ein Verfahren zur Umrechnung von Dezimalzahlen in Binärzahlen kennen gelernt. Erstellen Sie nun eine Methode, die eine ganze Zahl größer Null einliest und die entsprechende Binärdarstellung auf der Konsole ausgibt.

### 45 Potenzberechnung

- Schreiben Sie eine statische Methode, die die Potenz  $n^m$  (von ganzen positiven Zahlen) einzig und allein über die Multiplikation ( $2^4$  entspricht  $2*2*2*2$ ) berechnet und zurückliefert.
- Schreiben Sie eine weitere statische Methode, welche die Potenz allein durch Addition berechnet und zurückliefert ( $5^2 = 5*5 = 5+5+5+5+5$ ).
- Zusatz:* Schreiben Sie eine statische Methode, welche nur den ++-Operator nutzt, um die Potenz zu berechnen!

### 46 Rekursive Methoden

Implementieren Sie folgende Funktionalität rekursiv, d.h. die Verwendung von Schleifen ist verboten!

- `int getAnzahlZiffern(int zahl):`  
liefert die Anzahl an Ziffern der übergebenen Zahl,  
z.B. `getAnzahlZiffern(2542) → 4`
- `int getZiffernWert(int zahl, int stelle):`  
liefert den Wert der Ziffer der übergebenen Zahl an der Stelle; die Stellen werden dabei von rechts nach links angegeben und beginnen bei 0,  
z.B. `getZiffernWert(27381, 3) → 7` oder `getZiffernWert(27381, 0) → 1`
- `int ersetzeZiffer(int zahl, int stelle, int wert):`  
ersetzt die Ziffer der übergebenen Zahl an der Stelle durch den übergebenen Wert und liefert die neue Zahl; die Stellen werden dabei von rechts nach links angegeben und beginnen bei 0, z.B. `ersetzeZiffer(24135, 3, 7) → 27135`

## 47 Größter gemeinsamer Teiler

Der größte gemeinsame Teiler (ggT) zweier Zahlen  $a$  und  $b$  ist eine natürliche Zahl  $t$  mit folgenden Eigenschaften:

1.  $t$  teilt  $a$
2.  $t$  teilt  $b$

Jede andere Zahl, die  $a$  und  $b$  teilt ist auch automatisch Teiler von  $t$

**Beispiel:**  $\text{ggT}(18, 12) = 6$

(6 teilt 18 und 12, jede andere Zahl die 18 und 12 teilt – 1, 2 und 3 – ist Teiler von 6)

Schreiben Sie eine statische rekursive Methode um den ggT von zwei ganzen Zahlen zu berechnen.

Hierbei sind zwei Informationen von Vorteil:

$$\text{ggT}(0, x) = x$$

$$\text{ggT}(x, 0) = x$$

## 48 Zahlenfolge

Wir betrachten einmal die folgende Zahlenfolge:

0, 1, 3, 8, 20, 49, 119, ...

Dabei ergibt sich eine Zahl aus der Summe vom verdoppelten Vorgänger sowie dem um eins erhöhten Vorvorgänger.

- a) Welche beiden Zahlen folgen der 119?
- b) Schreiben Sie eine rekursive Methode, welche die  $n$ -te Zahl berechnet.
- c) Schreiben Sie eine iterative Methode, welche ebenfalls die  $n$ -te Zahl berechnet.

## 49 Fibonacci-Reihe

Die „Fibonacci-Reihe“ ist eine Folge von ganzen, positiven Zahlen  $f_0, f_1, f_2, \dots$ . Die ersten beiden Fibonacci-Zahlen sind  $f_0 = 0$  und  $f_1 = 1$ . Jede weitere Zahl ist die Summe der beiden Vorgänger:

$$f_n = f_{n-2} + f_{n-1} \quad \forall n \geq 2$$

Der Anfang der Fibonacci-Reihe lautet 0, 1, 1, 2, 3, 5, 8, ...

Schreiben Sie eine **rekursive** Funktion `fibonacci()`, die eine Zahl größer 0 akzeptiert und die entsprechende Fibonacci-Zahl  $f_n$  berechnet. Ignorieren Sie arithmetischen Überlauf.

## Weitere Aufgaben: Reihungen

### 50 Gleichheit von Arrays

Schreiben Sie eine statische Methode, welche zwei `int`-Arrays daraufhin vergleicht ob sie an jeder Stelle die gleichen Werte enthalten. Das Ergebnis der Prüfung wird der aufrufenden Methode zurückgegeben. Betrachten Sie bitte den Fall zwei Arrays unterschiedlicher Länge.

### 51 Array to Array

Schreiben Sie eine statische Methode, welche ein neues zweidimensionales Array in Abhängigkeit eines vorhandenen alten Arrays erstellt. Folgender Algorithmus soll hierbei Anwendung finden:

Die erste Zeile soll jeweils die Ergebnisse von den aktuellen Werten des alten Arrays modulo 1 beinhalten.

Die zweite Zeile soll jeweils die Ergebnisse von den aktuellen Werten des alten Arrays modulo 2 beinhalten. Die dritte Zeile ...

<i>Beispiel:</i> OriginalArray					<i>Beispiel:</i> ArrayNeu				
	1	2	3	4		0	0	0	0
	5	6	7	8		1	0	1	0
	9	10	11	12		0	1	2	0

*Zur Übung:*

Schreiben Sie zunächst eine statische Methode `fillArray`, welche das Array wie im `OriginalArray` belegt, sowie eine Methode `printArray`, welche das Array formatiert ausgibt.

### 52 int Arrays

Definieren Sie in der `main`-Methode ein `int`-Array fester Größe. Dieses Array dient nun als Eingabeparameter für die im Folgenden beschriebenen Unterprogramme:

- a) `fillArray()` liest von der Konsole Werte in das Array ein.
- b) `printArray()` gibt die Werte des Arrays auf der Konsole in ansprechender Form aus.
- c) `getSumme()` gibt die Summe der im Array abgelegten Werte zurück.
- d) `getMax()` gibt das größte Element des Arrays zurück.

## 53 Arrays sortieren

Definieren Sie in der main-Methode ein int-Array, dessen Größe Sie von der Konsole einlesen. Dieses Array dient nun als Eingabeparameter für die im Folgenden beschriebenen Methoden:

- a) `setFirstMax()` vertauscht im Array den Wert der ersten Komponente mit dem höchsten Wert, so dass der höchste Wert am Ende vorne steht.
- b) `reverseArray()` kehrt die Reihenfolge der Komponenten um. (Das letzte Element wird das erste, das vorletzte das zweite usw.)
- c) `sortArray()` sortiert das Array in aufsteigender Reihenfolge. (Dazu können Sie möglicherweise den Algorithmus aus Aufgabe a) verwenden. Das Verfahren wird üblicherweise mit "Selection – Sort" bezeichnet.)
- d) `insertSort()` belegt das Array mit Nullen vor und liest dann von der Konsole positive Zahlen ein. Diese Zahlen sollen in absteigender Reihenfolge sortiert werden. Dazu wird jede Zahl nach dem Einlesen an der richtigen Stelle einsortiert. Die Anzahl der eingelesenen Zahlen muss nicht mit der Größe des Arrays übereinstimmen. Werden weniger Zahlen eingelesen, bleiben die letzten Array – Elemente mit Nullen belegt, werden mehr eingelesen, werden die kleinsten Elemente hinausgeschoben. In diesem Fall ist allerdings eine Warnmeldung auf der Konsole auszugeben. (Das Verfahren wird üblicherweise mit "Insertion – Sort" bezeichnet.)

## 54 Zweidimensionales Array

Erstellen Sie ein Programm, welches die durchschnittlichen Verbräuche (Kraftstoff) eines Fuhrunternehmens (10 Fahrzeuge) widerspiegelt. Als Struktur wird ein zweidimensionales Array vorgegeben (`int[FahrzeugID][Monat]`). Schreiben Sie eine Funktion, welche alle Monatswerte zufällig belegt und eine weitere Funktion, welche die Quartalsverbräuche ausgibt.

## 55 Zahlenfolgen

Diese Aufgabe hat den Zweck Zahlenreihen zusammen zu fassen.

Schreiben Sie eine statische Methode `readNumbers()`, welche nach der Anzahl der Zahlen fragt, die Zahlen einzeln abfragt und die Zahlen aufsteigend sortiert als Array zurück liefert. Sie haben hier die Möglichkeit, ihre Funktion `InsertionSort()` gleich in der Eingabe oder den `SelectionSort()` nach der kompletten Eingabe zu benutzen.

Die sortierten Zahlenreihen sollen nun auf zwei verschiedene Arten ausgegeben werden.

- In der ersten Ausgabe werden alle Zahlen einzeln durch Komma getrennt ausgegeben. Schreiben Sie dazu die statische Methode `printCommaSeparatedInts()`, welche ein `int`-Array entgegen nimmt und die Zahlen durch Komma getrennt ausgibt.
- In der zweiten Ausgabe wird die Zahlenreihe verkürzt ausgegeben, sprich alle aufsteigenden zusammenhängenden Zahlenfolgen werden durch einen Bindestrich zwischen dem kleinsten und dem größten Wert der zusammenhängenden Reihe ausgegeben. Zwischen den einzelnen Zahlen bzw. Zahlenreihen erscheint wieder ein Komma. Schreiben Sie eine statische Methode `printCommaSeparatedIntsShort()`, welche ein `int`-Array entgegen nimmt und die Anforderungen unter b) erfüllt.

*Tipps:*

- Sollten Sie den Algorithmus zum Sortieren noch nicht implementiert haben, können Sie die Zahlen gleich sortiert eingeben.
- Wir gehen zunächst davon aus, dass der Nutzer jede Zahl nur einmal verwendet.
- `printCommaSeparatedIntsShort()`: Zunächst keine Kommas betrachten! Nachdem die Zusammenfassung erfolgreich ist, Kommas einfügen.

*Beispiel:*

<b>Eingabe:</b> 1 4 3 5 8 10 11 6 <b>Ausgabe1:</b> „1,3,4,5,6,8,10,11“ <b>Ausgabe2:</b> „1,3-6,8,10-11“
--

*Zusatz:*

Fangen Sie den Fall ab, bei dem der Nutzer eine Zahl mehrmals eingeben möchte. Wenn der Nutzer dies versucht, soll er darauf hingewiesen werden und eine erneute Eingabe vornehmen. Die Prüfung auf doppelte Eingabe kann auch in eine eigene Methode ausgelagert werden.

Testen Sie ihr Programm nach jeder Änderung!

## 56 Primfaktorzerlegung

Bei der Primfaktorzerlegung wird eine Zahl als das Produkt ihrer Primfaktoren, also ein Produkt aus Primzahlen dargestellt. Zum Beispiel kann man 12 als  $2 \cdot 2 \cdot 3$  schreiben oder 16 als  $2 \cdot 2 \cdot 2 \cdot 2$ . Dabei heißen die einzelnen Faktoren, aus denen das Produkt besteht, Primfaktoren. Die Primfaktordarstellung einer Zahl ist bis auf ihre Reihenfolge der Primfaktoren eindeutig.

Schreiben Sie eine Funktion, welche zu einer gegebenen Zahl, ihre Primfaktoren ausgibt:

*Beispiele:*

Eingabe:

3

Primfaktoren: 3

Eingabe:

12

Primfaktoren:  $2 \cdot 2 \cdot 3$

Eingabe:

12458

Primfaktoren:  $2 \cdot 6229$

## 57 EURO Seriennummern

Euro-Banknoten haben eine eindeutige Seriennummer, die aus einem führenden Buchstaben, 10 Ziffern und einer Prüfziffer bestehen.

Beispiel: Z 6016220022 6 (Leerzeichen zur Übersicht hinzugefügt).

Der führende Buchstabe codiert die nationale Zentralbank (NZB), die den Geldschein im Umlauf gebracht hat. Sie wird NZB-Nummer genannt.

Die Prüfziffer berechnet sich wie folgt:

- Der Buchstabe wird durch seine Position im lateinischen Alphabet ersetzt (bei Z also 26)
- Es wird die Quersumme der Positionszahl und der 10 Ziffern berechnet (im Beispiel  $2+6 + 6+0+1+6+2+2+0+0+2+2 = 29$ )
- Die Zahl wird mit Rest durch 9 geteilt ( $29 : 9 = 3$  Rest 2).
- Der Rest wird von 8 subtrahiert ( $8 - 2 = 6$ ). Das Resultat ist die Prüfziffer (6). Falls die Prüfziffer 0 ist, wird sie durch 9 ersetzt.

Implementieren Sie eine statische Methode, die für eine Seriennummer (gegeben als char-Array `seriennummer`) die Prüfziffer berechnet und zurückgibt.

Format:

`seriennummer[0]` ist der Buchstabe, `seriennummer[11]` die Prüfziffer. Der Buchstabe ist immer groß. Fehlerfälle brauchen nicht berücksichtigt werden.