

Übungsblatt 7 Java

1. Tabelle von Zweierpotenzen

Erstellen Sie ein Java Programm, das alle positiven und negativen Potenzen zu 2 in Form einer Tabelle (bis Potenz 30) ausgibt.

Tabelle für Zweierpotenzen

2 hoch i	i	2 hoch -i
1	0	1
2	1	0,5
4	2	0,25
8	3	0,125
16	4	0,0625
32	5	0,03125
64	6	0,015625
128	7	0,0078125
256	8	0,00390625
512	9	0,001953125
1 kilobyte		
1024	10	0,0009765625
2048	11	0,00048828125
4096	12	0,000244140625
8192	13	0,0001220703125
16384	14	0,00006103515625
32768	15	0,000030517578125
65536	16	0,0000152587890625
131072	17	0,00000762939453125
262144	18	0,000003814697265625
524288	19	0,0000019073486328125
1 Megabyte		
1048576	20	0,00000095367431640625
2097152	21	0,000000476837158203125
4194304	22	0,0000002384185791015625
8388608	23	0,00000011920928955078125
16777216	24	0,000000059604644775390625
33554432	25	0,000000029802322387695312
67108864	26	0,000000014901161193847656
134217728	27	0,000000007450580596923828
268435456	28	0,000000003725290298461914
536870912	29	0,000000001862645149230957
1 Gigabyte		
1073741824	30	0,0000000009313225746154785

Hinweis: Verwenden Sie die statischen Methoden `Console.padStringLeft()` für die linksbündige Ausgabe einer Ganzzahl und `Console.Double2String()` für die Formatierung der Kommazahlen. Der Formatstring sei "0.#####".

2. Summe von Ganzen Zahlen

Schreiben Sie eine Java Klasse mit Namen `Summe`, die die Zahlen von einem einzugebenden Anfangswert bis zu einem einzugebenden Endwert in einem Unterprogramm aufsummiert und das Ergebnis ausgibt.

$$summe = \sum_{i=a}^e i = a + (a+1) + (a+2) + \dots + e$$

a : Anfangswert *e* : Endwert

Programm zur Berechnung der Summe aller Zahlen zwischen Anfangswert und Endwert.

Bitte geben Sie den Anfangswert ein: 1

Bitte geben Sie den Endwert ein: 100

Die Summe der Zahlen von 1 bis 100 = 5050

3. String rückwärts

Erstellen Sie ein Java Programm mit Namen Reverse, das den Inhalt einer Textvariablen von hinten her liest.

- Schreiben sie ein Unterprogramm `stringReverse(String):String` zur iterativen Lösung (mit Schleifen) des Problems.
- Schreiben Sie ein Unterprogramm `arrayReverse(char[], int)` zur rekursiven (das Unterprogramm ruft sich selbst auf) Lösung.

Doppeltes Aufrufen des Unterprogramms erzeugt wieder das Original.

miehsniS - EIGT
TGIE - Sinsheim

4. Bubblesort

Sortieren Sie ein Array von 10 zufälligen Ganzzahlen im Bereich von 0 bis 100 der Größe nach absteigend. Ein einfacher Algorithmus geht mehrfach durch das ganze Array und vertauscht benachbarte Elemente, wenn das nachfolgende Element größer als das aktuelle ist. Das Sortieren ist beendet wenn keine Vertauschungen mehr stattfinden.

66 53 60 21 83 60 95 6 44 15
95 83 66 60 60 53 44 21 15 6

Vergleichen Sie Ihre Implementierung mit der Bibliotheksfunktion `Arrays.sort()` hinsichtlich der Laufzeit. Um die Laufzeit zu messen, bietet sich `start = System.nanoTime();` an.

5. Text formatieren

Schreiben Sie ein Programm zur linksbündigen, rechtsbündigen und zentrierten Ausgabe einer Textdatei. Verwenden Sie zum einlesen der Datei die Klasse `FileInput.readFile(String): String[]`. Legen Sie die maximale Zeilenlänge in Ihrem Programm als Konstante fest.

z.B.

	1	2	3	4	5	6	7
	-----	-----	-----	-----	-----	-----	-----
	Stufen						
	<p>wie jede Blüte welkt und jede Jugend Dem Alter weicht, blüht jede Lebensstufe, Blüht jede Weisheit auch und jede Tugend zu ihrer Zeit und darf nicht ewig dauern. Es muß das Herz bei jedem Lebensrufe Bereit zum Abschied sein und Neubeginne, Um sich in Tapferkeit und ohne Trauern in andre, neue Bindungen zu geben. Und jedem Anfang wohnt ein Zauber inne, Der uns beschützt und der uns hilft zu leben.</p> <p>wir sollen heiter Raum um Raum durchschreiten, An keinem wie an einer Heimat hängen, Der Weltgeist will nicht fesseln uns und engen, Er will uns stuf' um Stufe heben, weiten. Kaum sind wir heimisch einem Lebenskreise Und traulich eingewohnt, so droht Erschlaffen, Nur wer bereit zu Aufbruch ist und Reise, Mag lähmender Gewöhnung sich entrafen.</p> <p>Es wird vielleicht auch noch die Todesstunde Uns neuen Räumen jung entsendend, Des Lebens Ruf an uns wird niemals enden ... Wohlan denn, Herz, nimm Abschied und gesunde! (Hermann Hesse)</p>						

Übungsblatt 8 Java

7. Sortieren von Text

a.) Erstellen Sie ein Java Programm, das drei in einem Array gespeicherte Namen alphabetisch sortiert.

b.) Erweitern Sie Ihr Programm so, dass die Daten aus einer Textdatei gelesen werden können (addr.txt oder addrKurz.txt). Verwenden Sie die Klasse FileInput. Der Benutzer soll eine Spalte als Sortierkriterium wählen können und dann diese sortiert angezeigt bekommen.

Ausgabe des Programms:

```
0  Vorname
1  Name
2  Strasse
3  Haus Nr.
4  PLZ
5  Ort
6  Tel.
7  e-Mail
```

Wählen Sie ein Sortierkriterium: 0

Vorname	Name	Strasse
Dalibor	Colakovic	Zwickauer Weg
Daniel	Steudel	Am Hohenstein
David	Beckmann	Rosenhofstr.
...		
Sascha	Jäger	Ringstr.
Stefanie	Reiss	Kreuzäcker
Tom	Welker	Hauptstr.

8. Schnittpunkt zweier Geraden in der Ebene

Schreiben Sie eine Java Klasse mit Namen Gerade, die den Schnittpunkt zweier Geraden berechnet. Als Eingabe sei für jede Gerade die Steigung und der Schnittpunkt mit der Y-Achse gegeben.

9. Schnittpunkte zweier Kreise in der Ebene

Erstellen Sie ein Java Programm mit Namen Kreis2, das die Schnittpunkte zweier Kreise in der Ebene berechnet. Für jeden Kreis sind der Mittelpunkt und der Radius gegeben.

Die allgemeine Gleichung zur Beschreibung eines Kreises mit dem Mittelpunkt $M(x_1, y_1)$ und dem Radius r_1 lautet:

$$(x - x_1)^2 + (y - y_1)^2 = r_1^2$$

Kreis 1 $m_1 = (3.0, 5.0)$ $r_1 = 5.0$

Kreis 2 $m_2 = (-1.0, 1.0)$ $r_2 = 1.0$

1. Schnittpunkt: $x = 0.0$ $y = 1.0$

2. Schnittpunkt: $x = -1.0$ $y = 2.0$

Übungsblatt 9 Java

1. Ausgabe

Was machen die folgenden Programmzeilen?

- a. `System.out.println("3 + 4");`
- b. `System.out.println(3 + 4);`
- c. `System.out.println(3 + "4");`

2. Name

Schreiben Sie ein Programm zur Anzeige ihres Namens mit Rahmen auf dem Bildschirm.

```
Geben Sie Ihren Namen ein: Joachim Heitel
+-----+
|Joachim Heitel|
+-----+
```

3. Gesicht

Schreiben Sie ein Programm zur Ausgabe des folgenden Gesichts. Vielleicht zeichnen Sie noch ein schöneres.

```
  // // //
  |  o  o  |
  |  ^  |
  | [ ] |
  |_____|
  -----
```

4. Tic-Tac-Toe

Schreiben Sie ein Programm das untenstehendes Tic-Tac-Toe-Spielfeld ausgibt.

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

Es soll nur **eine** Stelle geändert werden, wenn man ein 4x4 oder 5x5 Spielfeld haben möchte.

5. Treppe

Schreiben Sie ein Programm das untenstehende Treppe ausgibt.

```

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

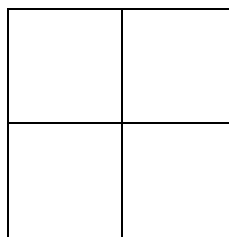
Es soll nur **eine** Stelle geändert werden, wenn man eine höhere Treppe haben möchte.

6. Rechteck

- a) Schreiben Sie ein Programm, in dem ein Objekt vom Typ Rectangle angelegt wird. Geben Sie das Objekt aus. Berechnen Sie zu diesem Object den Umfangs und die Fläche! (Nutzen sie die Methoden getWidth() und getHeight()).

```
java.awt.Rectangle[x=10,y=10,width=30,height=40]
Umfang: 140.0
Fläche: 1200.0
```

- b) Schreiben Sie ein Programm, in dem ein Objekt vom Typ Rectangle angelegt wird. Erzeugen Sie drei Kopien des Objekts (Methode clone()). Verschieben Sie die Rechtecke so dass wenn man sie zeichnen würde folgendes Bild entsteht:



Verwenden Sie die Methode translate().

- c) Schreiben Sie ein Programm, in dem zwei Objekte vom Typ Rectangle angelegt werden. Berechnen Sie die Schnittfläche mit Hilfe der Methode intersection(). Was passiert wenn die beiden Rechtecke keine gemeinsame Fläche haben?

```
r1 = java.awt.Rectangle[x=10,y=10,width=30,height=40]
r2 = java.awt.Rectangle[x=20,y=20,width=40,height=50]
r3 = java.awt.Rectangle[x=20,y=20,width=20,height=30]
```

7. Farben

Schreiben Sie ein Programm, in dem ein Objekt vom Typ Color angelegt wird. Initialisieren Sie dieses Objekt mit den Werten 50, 100, 150. Wenden Sie die Methode brighter() an. Welche Auswirkung hat sie? Wiederholen Sie das Experiment mit der Methode darker().

8. Buchstaben

- a) Schreiben Sie ein Programm, das eine Zeile Text einliest und alle auftretenden i mit ! und alle s mit \$ Zeichen ersetzt. Prüfen Sie ihr Programm mit dem Wort Mississippi.
- b) Schreiben Sie ein Programm, das alle e und o in einem Text vertauscht. Aus „Hello World!“ wird „Holle Werld!“

9. Kalender

Lesen Sie sich in der Java Hilfe die Dokumentation der Klasse GregorianCalendar durch.

- a) Schreiben Sie ein Programm, das Datum und den Wochentag von heute in 100 Tagen ausgibt. (Hinweis: Methode add(), Lese Kapitel 10 in „Java ist auch eine Insel“, siehe auch die Klasse SimpleDateFormat)
- b) Berechnen Sie den Wochentag ihres Geburtstages. Verwenden Sie Outlook, Excel oder einen anderen Kalender zur Prüfung.
- c) Wie alt sind Sie in Tagen? Geben Sie Ihr Alter auch in Jahren, Monaten und Tagen aus.

```
Heute ist: Freitag, der 03.März.2006
Geburtstag am: Dienstag, der 23.März.1965
```

```
Alter in Tagen: 14955
```

```
Jahre: 40
Monate: 11
Tage: 11
```

- d) Schreiben Sie eine Klasse Kalenderd, die wenn man sie ohne Parameter startet, den Kalender des aktuellen Monats und des aktuellen Jahres ausgibt.

```
D:\Java\workspace\kalender>java kalenderd
März 2006
Mo Di Mi Do Fr Sa So | woche
    1 2 3 4 5 | 9
 6 7 8 9 10 11 12 | 10
13 14 15 16 17 18 19 | 11
20 21 22 23 24 25 26 | 12
27 28 29 30 31
```

Wenn man eine Zahl auf der Kommandozeile ergänzt, so kann man den Monat des aktuellen Jahres wählen z. B.:

```
D:\Java\workspace\kalender>java kalenderd 5
Mai 2006
Mo Di Mi Do Fr Sa So | woche
 1 2 3 4 5 6 7 | 18
 8 9 10 11 12 13 14 | 19
15 16 17 18 19 20 21 | 20
22 23 24 25 26 27 28 | 21
29 30 31
```

Gibt man zwei Zahlen auf der Kommandozeile, so kann man auch noch das Jahr wählen z. B.:

```
D:\Java\workspace\kalender>java kalenderd 12 2000
Dezember 2000
Mo Di Mi Do Fr Sa So | woche
    1 2 3 | 48
 4 5 6 7 8 9 10 | 49
11 12 13 14 15 16 17 | 50
18 19 20 21 22 23 24 | 51
25 26 27 28 29 30 31 | 52
```

Geben Sie folgende Befehle in eine Textdatei mit Namen cal.bat. Rufen sie cal auf. Machen Sie sich die Wirkungsweise der Befehle bewusst.

```
echo off
chcp 1252
set i=1

:MARKE
if %i%==13 goto ENDE
java Kalenderd %i% %1
set /a i+=1
goto MARKE

:ENDE
```

10. Fehlersuche

Finden Sie die Fehler in den folgenden Programmzeilen. Handelt es sich um Laufzeit oder Compilerfehler?

- a. `Rectangle r = (5, 10, 15, 20);`
- b. `double width = Rectangle(5, 10, 15, 20).getWidth();`
- c. `Rectangle r;`
`r.translate(15, 25);`
- d. `r = new Rectangle();`
`r.translate("far, far away!");`

11. Overload

Finden Sie mehrere überladene Methoden der Klasse

- a) `String`
- b) `Rectangle`

Übungsblatt 10 Java

1. Größter gemeinsamer Teiler

Schreiben Sie ein Unterprogramm zur Berechnung des größten gemeinsamen Teilers zweier Zahlen (ggT wird benötigt beim Kürzen von Brüchen). Das Hauptprogramm soll folgende Bildschirmansicht erzeugen.

```

Programm zur Berechnung des größten gemeinsamen Teilers zweier Zahlen z1
und z2. (ggT)
Bitte geben Sie z1 ein: -32
Bitte geben Sie z2 ein: 24
Der größte gemeinsame Teiler von -32 und 24 ist: 8
  
```

2. Multiplikationstabelle

Schreiben Sie ein Programm zur Berechnung und Anzeige einer Multiplikationstabelle.

```

Programm zur Anzeige einer Multiplikationstabelle:
~~~~~
wie groß soll die Tabelle werden? 5
 1  2  3  4  5
 2  4  6  8 10
 3  6  9 12 15
 4  8 12 16 20
 5 10 15 20 25
  
```

3. Verschiebeoperatoren

Lesen Sie in Java ist auch eine Insel Kapitel 2.9.4 über die Verschiebeoperatoren nach. Weisen Sie nach, dass folgende Ausdrücke das gleiche Ergebnis liefern.

`x%8*4` entspricht `(x-(x>>3 << 3))<<2`

Spalten Sie den rechten Ausdruck in Teilausdrücke auf und lassen Sie sich diese Dual anzeigen.

4. Runden

Schreiben Sie ein statisches Unterprogramm „runden“, das eine Fließkommazahl als Rundungswert und eine Ganzzahl für die Anzahl der Stellen übergeben bekommt. Ist die Anzahl der Stellen negativ so soll auf 10er, 100er oder 1000er gerundet werden, ist die Anzahl der Stellen positiv so soll auf zehntel oder hundertstel gerundet werden. Verwenden Sie als Hilfsfunktion die Funktion Math.floor().

```
Aufruf runden(3.141592653589793, 5) ergibt
3.14159
Aufruf runden(123456.0, -3) ergibt
123000.0
```

5. Zeitberechnung

Schreiben Sie ein Java Programm, das eine Anzahl von Sekunden einliest und in Tagen Stunden, Minuten und Sekunden umrechnet.

```
Umrechnung von Sekunden in Tage Stunden:Minuten:Sekunden
-----
Sekunden: 12345
Tage: 0 03:25:45
```

6. Polynomberechnung

Schreiben Sie ein Java Programm, das die Koeffizienten $a_5 - a_0$ als Array mit Initialisierung anlegt und den Wert des Polynoms für $0 \leq x \leq 1000$ in hundertstel Schritten berechnet und in einem Array zwischen speichert. Am Schluss soll eine Wertetabelle angezeigt werden.

$$f(x) = a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0$$

```
Berechnung der Wertetabelle der Funktion:
f(x) = 5.0 * x^4 + 4.0 * x^3 + 3.0 * x^2 + 2.0 * x^1 + 1.0
0.0    1.0
0.01   1.02030405
0.02   1.0412328
...
999.96  5.003202569778846E12
999.97  5.003402668831203E12
999.98  5.003602773885601E12
999.99  5.00380288494216E12
```

Schreiben Sie eine Version unter Verwendung der Mathematikbibliotheksfunktion Math.pow und eine Version unter Verwendung des Hornerschemas. Vergleichen sie die Rechenzeiten. (HornerSchema bei Wikipedia nachlesen!).

7. Schatzsuche

Auf der Landkarte eines Schatzsuchers ist vermerkt, dass sich ein Schatz bei der geografischen Breite von $49,2586^\circ$ (nördlich) und der geografischen Länge von $8,8786^\circ$ (östlich) befindet.

- Informieren Sie sich bei Wikipedia über den Begriff Bogenminute und Bogensekunde.
- Rechnen Sie obige Koordinaten von Hand in die Form Grad/Minute/Sekunde um.
- Schreiben Sie ein Programm zur Umrechnung der Winkeldarstellung in beide Richtungen.
- Suchen Sie bei GoogleEarth den Schatz!

