

Aufgabenblatt 15

Zusätzliche Aufgaben

keine Abgabe

*Vorlesung Programmieren
Prof. Dr. Dirk Eisenbiegler
Hochschule Furtwangen*

Aufgabe 15.1 - Zeichensatz ausgeben

In dieser Aufgabe soll der Zeichensatz in Tabellenform ausgeben werden.

Schreiben Sie eine Methode, die für jede Zahl x aus einem Zahlenbereich zwischen a und b jeweils eine Zeile ausgibt. In jeder Zeile sollen nacheinander die folgenden Daten ausgegeben werden:

- ⇒ die Zahl x selbst
- ⇒ der Buchstabe, dessen Code x ist
- ⇒ x dargestellt als Hexadezimalzahl
- ⇒ x dargestellt als Binärzahl

a und b sind Parameter der zu implementierenden Methode. Beide sind vom Typ `int`.

Hinweise:

x mit dem Ausdruck

```
(char) x
```

wird der Buchstabe bestimmt, dessen Code x ist.

x der Ausdruck

```
Integer.toString(x,b)
```

konvertiert die `int`-Zahl x in einen `String`, in dem der Wert der Zahl im Zahlensystem mit der Basis b dargestellt wird.

Umwandlung in Binärdarstellung: $b=2$.

Umwandlung in Hexadezimaldarstellung: $b=16$.

Aufgabe 15.2 - Quadrate

Schreiben Sie eine Methode *quadrat*, die Quadrate auf den Bildschirm zeichnet. Die Kanten der Quadrate bestehen aus "*" -Zeichen. Die Quadrate sollen mit "-" -Zeichen ausgefüllt werden. Die Methode *quadrat* soll beliebig große Quadrate zeichnen können. Die Kantenlänge *k* ist Parameter der Methode *quadrat*. Die nachfolgenden Beispiele zeigen Quadrate mit k=4, k=7 und k=1.

```
****
*--*
*--*
****
```

```
*****
*-----*
*-----*
*-----*
*-----*
*-----*
*****
```

```
*
```

Aufgabe 15.3 - Primzahlen

- A) Schreiben Sie eine Methode *teilbar*, die zu zwei int-Zahlen x und y bestimmt, ob x durch y teilbar ist. Der Rückgabewert ist vom Typ boolean.
Hinweis: Der Java-Ausdruck a%b bestimmt den Rest einer Division von a und b.
- B) Jede Zahl lässt sich in eindeutiger Weise in ein Produkt von Primzahlen zerlegen. Beispiel: 168=2·2·2·3·7
Schreiben Sie eine Methode *primfaktoren*, die zu einer int-Zahl die Primfaktorzerlegung ausgibt.
- C) Schreiben Sie eine Methode *prim*, die bestimmt ob eine int-Zahl eine Primzahl ist. Der Parameter der Zahl prim hat den Typ int, der Rückgabewert ist vom Typ boolean.
Primzahlen sind alle Zahlen, die nur durch 1 und durch sich selbst teilbar sind. Die Zahl 1 ist keine Primzahl.
Beispiele für Primzahlen: 2, 3, 5, 7, 11, 13, 17, 19,...
- D) Schreiben Sie eine Methode, die zu einem bestimmten Intervall [1:n] bestimmt, welche der darin enthaltenen Zahlen Primzahlen sind. Verwenden Sie dazu das Verfahren „Sieb des Erathosthenes“. https://de.wikipedia.org/wiki/Sieb_des_Eratosthenes

Aufgabe 15.4 - Rechnen mit Vektoren, kartesische Koordinaten

In dieser Aufgabe sollen dreidimensionale Vektoren durch double-Arrays der Länge drei repräsentiert werden.

Schreiben Sie statische Methoden zur Addition und Subtraktion von Arrays. Schreiben Sie eine Methode zur Berechnung der Skalarmultiplikation und eine Methode zur Berechnung des Skalarprodukts.

Eine quadratische 3x3-Matrix soll durch einen zweidimensionalen double-Array repräsentiert werden. Schreiben Sie eine Methode zur Multiplikation einer Matrix mit einem Vektor! Schreiben Sie eine Methode zur Multiplikation zweier quadratischer 3x3-Matrizen.

Aufgabe 15.5 - Sortieren

Schreiben Sie eine Methode, die einen Array von double-Werten sortiert.

Hinweise:

- x Die Methode hat einen Parameter vom Typ double-Array (eine Objekt-Referenz!). Der Rückgabewert der Methode ist void. Innerhalb der Methode sollen die Variablenwerte an den einzelnen Positionen so vertauscht werden, dass schließlich alle Werte in aufsteigender Reihenfolge sortiert sind.
- x Einfacher Algorithmus (als Vorschlag):
 - ⇒ Das Minimum aller Werte suchen, diesen Wert mit dem Wert an der ersten Position vertauschen.
 - ⇒ Ab der zweiten Position nach dem minimalen Wert suchen, diesen Wert mit dem Wert an der zweiten Position vertauschen.
- x Weitere Sortieralgorithmen finden Sie auf Wikipedia:
<https://de.wikipedia.org/wiki/Sortierverfahren>
 - ⇒ Vorschlag: Implementieren Sie Bubblesort, Quicksort und Mergesort.

Aufgabe 15.6 - Morsen

Schreiben Sie eine Methode f , die einen Text in Morsezeichen übersetzt und eine Methode g für die Rückübersetzung der Morsezeichen in eine Buchstabenfolge.

Der Text bestehe aus den Großbuchstaben A bis Z und zusätzlich aus Leerzeichen. Beispiel:

MORGEN WIRD ES REGNEN

Sowohl der Text als auch die Kodierung mit Morsezeichen soll jeweils durch einen String repräsentiert werden. Die Methoden f und g haben je einen String als Parameter und einen String als Rückgabewert.

Die Methode f soll diesen Text in Morsezeichen übersetzen. Ein Morsezeichen besteht aus mehreren Punkten und Strichen gemäß nachfolgender Tabelle. Ein Punkt steht für einen kurzen Ton, ein Strich für einen langen Ton. Als Trennzeichen zwischen zwei Morsezeichen soll ein Leerzeichen verwendet werden. Ein Leerzeichen steht für eine kurze Pause. Zwischen zwei Worten sollen im Morsecode zwei aufeinanderfolgende Leerzeichen stehen.

Die Methode g soll die Morsekodierung zurück in den Text übersetzen.

A	.-	B	-...	C	-. -.
D	-..	E	.	F	..-.
G	--.	H	I	..
J	.---	K	-. -	L	.-..
M	--	N	-. .	O	---
P	.-. .	Q	--. -	R	.-.
S	...	T	-	U	..-
V	...-	W	.- -	X	-. -
Y	-. - -	Z	--..		

Aufgabe 15.7 - Nullstellensuche durch Intervallschachtelung

Schreiben Sie zwei Methoden mit den Namen f und $nullstelle$.

Die Methode f soll die Funktion $f(x)=e^x+x^2-4$ realisieren. Der Parameter und der Rückgabewert der Funktion f sollen beide vom Typ *double* sein.

Die Methode $nullstelle$ soll in einem Intervall $[x,y]$ per Intervallschachtelung nach einer Nullstelle suchen. Das Iterationsverfahren soll erst dann abbrechen, wenn ein Wert m gefunden wird, sodass $|f(m)|<z$ gilt. Ist ein solcher Wert m gefunden, so soll die Methode diesen Wert zurückgeben. Die Werte x , y und z sind Parameter der Methode $nullstelle$. Alle Parameter und der Rückgabewert von $nullstelle$ seien vom Typ *double*. Implementieren Sie die Methode $nullstelle$ mit Hilfe von Rekursion.

Testen Sie die Methode $nullstelle$ mit den Werten $x=-1.0$, $y=20.0$, $z=0.0001$ sowie mit den Werten $x=-100.0$, $y=0.0$, $z=0.0001$.

Hinweise:

\times Die Java-Funktion $\text{Math.exp}(x)$ berechnet e^x .

\times Die Java-Funktion $\text{Math.abs}(x)$ berechnet $|x|$.

Es darf angenommen werden, dass von den beiden Werten $f(x)$ und $f(y)$ einer der beiden Werte größer als 0 ist und der andere kleiner als 0.

Aufgabe 15.8 - Nullstellensuche durch Newton-Iteration

Schreiben Sie drei Methoden mit den Namen *f* , *g* und *nullstelle*.

Die Methode *f* soll die Funktion $f(x)=e^x+x^2-4$ und die Methode *g* die Funktion $g(x)=e^x+2\cdot x$ realisieren. Die Funktion *g* ist die erste Ableitung von *f*. Die Parameter und die Rückgabewerte der Funktionen *f* und *g* sollen vom Typ *double* sein.

Die Methode *nullstelle* soll per Newton-Iteration nach einer Nullstelle von *f* suchen. Parameter von *nullstelle* sind *x* und *z*. Der Wert *x* ist der Startwert der Iteration, der Wert *z* ist die Genauigkeit. Das Iterationsverfahren soll abbrechen, sobald ein Wert *m* gefunden wird, sodass $|f(m)|<z$ gilt.

Newton-Iteration

Von einem Näherungswert *x* zum nächsten Näherungswert *x'* kommt man wie folgt:

$$x' = x - \frac{f(x)}{f'(x)}$$

Hinweise:

- x* Die Java-Funktion *Math.exp(x)* berechnet e^x .
- x* Die Java-Funktion *Math.abs(x)* berechnet $|x|$.

Aufgabe 15.9 - Bitoperationen, RGB-Farbkodierung

In dieser Aufgabe wird die Klasse *Picture* verwendet, die in den regulären Aufgaben vorgestellt wird. Bitte beachten Sie die Erläuterungen auf diesem Aufgabenblatt.

In jeder der Teilaufgaben soll eine andere Operation an den Farben vorgenommen werden. Ändern Sie jeweils die Farben an den Punkten, die im Kreis mit dem Mittelpunkt (150,150) und dem Radius 100 liegen. Alle anderen Punkte sollen unverändert bleiben.

Jeder Punkt wird durch einen int-Wert dargestellt. Die ersten 8 Bit sind ungenutzt und müssen immer auf 1 gesetzt sein. In den nächsten 8 Bit befindet sich die Farbintensität für die Farbe Rot, in den nächsten 8 Bit die der Farbe Grün und in den letzten 8 Bit die Farbintensität für die Farbe Blau.

1	1	1	1	1	1	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0
255								32								7								128								
ungenutzt								Rot-Anteil								Grün-Anteil								Blau-Anteil								

- A) Setzen Sie in dem Kreis den Rot-Anteil auf konstant 128.
- B) Setzen Sie in dem Kreis für jede Farbe den Farbanteil von *x* auf 255-*x*.
- C) Vertauschen Sie die Farbanteile: der Rot-Anteile wird zum Grün-Anteil, der Grün-Anteil zum Blau-Anteil und der Blau-Anteil zum Rot-Anteil.

Hinweise:

- x* Durch die Operationen (*a* & *b*) werden zwei int-Zahlen *a* und *b* durch bitweises UND miteinander verknüpft. Das Ergebnis ist vom Typ int.
- x* Durch die Operationen (*a* | *b*) werden zwei int-Zahlen *a* und *b* durch bitweises ODER miteinander verknüpft. Das Ergebnis ist vom Typ int.
- x* Mit (*a* >> *b*) wird eine int-Zahl berechnet, die entsteht, wenn man die Bits im int-Wert *a* um *b* Stellen nach rechts verschiebt.
- x* Mit (*a* << *b*) wird eine int-Zahl berechnet, die entsteht, wenn man

die Bits im int-Wert a um b Stellen nach links verschiebt.

- x Um int-Zahlen im Hexadezimalsystem statt im Dezimalsystem anzugeben, muss lediglich 0x vorangestellt werden.

Beispiel: 0xFF4A3B87

Rot-Anteil: 4A, Grün-Anteil: 3B, Blau-Anteil: 87