

Aufgabenblatt 14

*Zusätzliche Aufgaben
keine Abgabe*

*Programmierung
Sommersemester 2015
Prof. Dr. Dirk Eisenbiegler
Hochschule Furtwangen*

Aufgabe 14.1 - Zeichen vertauscht ausgeben

- A) Schreiben Sie eine Methode, die die Zeichen eines String vertauscht ausgibt:
1. Zeichen, letztes Zeichen, 2. Zeichen vorletztes Zeichen, 3. Zeichen, ...

Beispiel

Parameter: Kokosnusspalme

Ausgabe: Keomkloaspnsus

- B) Schreiben Sie eine Methode, die die Zeichen eines String in folgender Reihenfolge ausgibt:
1. Zeichen, 3. Zeichen, 2. Zeichen, 4. Zeichen, 3. Zeichen, ...
Prinzip: zwei Schritte vor, einen Schritt zurück - bis der letzte Buchstabe erreicht wird

Beispiel

Parameter: Kokosnusspalme

Ausgabe: Kkooksonsunsusspsaplamle

Aufgabe 14.2 - Bitoperationen

In dieser Aufgabe wird die Klasse Picture verwendet, die bereits in Aufgabe 6.2 vorgestellt wurde.

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
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

Aufgabe 14.3 - 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 14.4 - 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 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 14.5 - 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 14.6 - 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) 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, ...
- C) Jede Zahl lässt sich in eindeutiger Weise in ein Produkt von Primzahlen zerlegen. Beispiel: $168=2\cdot2\cdot2\cdot3\cdot7$
Schreiben Sie eine Methode *primfaktoren*, die zu einer int-Zahl die Primfaktorzerlegung ausgibt.

Aufgabe 14.7 - Geheimschrift

Schreiben Sie zwei Methoden *encrypt* und *decrypt*, mit der ein String *s* verschlüsselt bzw. entschlüsselt werden soll.

Der String *s* soll Zeichen für Zeichen durch Vertauschung verschlüsselt werden. Wie vertauscht wird, wird durch einen Verschlüsselungsstring *x* beschrieben:

- ⇒ Ist das zu verschlüsselnde Zeichen in *x* enthalten, so soll es durch das nachfolgende Zeichen in *x* ersetzt werden.
- ⇒ Ist das zu verschlüsselnde Zeichen das letzte Zeichen in *x*, so wird es durch das erste Zeichen in *x* ersetzt.
- ⇒ Ist das Zeichen nicht in *x* enthalten, so bleibt es unverändert.

Beispiel:

s="Hallo Herr Müller!"

x="aKurHül!"

Verschlüsselter String: "üK!!o üeHH MI!!eHa"

Die Entschlüsselungsfunktion vertauscht genau entgegengesetzt.

Die Methode *encrypt* hat die beiden String-Parameter *s* und *x* und einen Rückgabewert vom Typ String. Die Methode *decrypt* hat ebenfalls zwei String-Parameter *s* und *x* und einen Rückgabewert vom Typ String.

Aufgabe 14.8 - Skalarprodukt

Schreiben Sie eine Methode, die das Skalarprodukt zweier Vektoren berechnet.

Die Methode hat zwei Parameter *a* und *b*. Beide Parameter sind int-Arrays. Die int-Arrays stehen für die zu multiplizierenden Vektoren. Es soll angenommen werden, dass beide Arrays die gleiche Länge haben. Der Rückgabewert der Methode sei vom Typ int und sein Wert sei das Skalarprodukt der Vektoren *a* und *b*.

Hinweis:

Unter dem (kanonischen) Skalarprodukt zweier Vektoren

$\vec{a}=(a_0,a_1,\dots,a_{n-1})$ und $\vec{b}=(b_0,b_1,\dots,b_{n-1})$ versteht man:

$$\vec{a} \cdot \vec{b} = a_0 \cdot b_0 + a_1 \cdot b_1 + \dots + a_{n-1} \cdot b_{n-1}$$

Aufgabe 14.9 - Sortieren

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

Hinweis:

- 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.
 - ⇒ ...

Aufgabe 14.10 - Morsen

In dieser Aufgabe soll der Roboter Morsezeichen erzeugen.

Beim Morsen gibt es zwei verschiedene Signalwerte: Strich und Punkt. Ein Strich dauert 0,3s, ein Punkt dauert 0,1s. Der Roboter soll Striche und Punkte durch das gleichzeitige Einschalten der beiden Lämpchen und durch einen Ton signalisieren – jeweils mit der entsprechenden Länge. Der Ton soll bei einem Strich die Frequenz 440Hz und bei einem Punkt die Frequenz 659Hz haben. Zwischen den Zeichen soll jeweils eine Pause von 0,1s sein, zwischen zwei Buchstaben eine Pause von 0,3s.

Morse-Alphabet:

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	---..		

Der Roboter soll die folgende Buchstabenfolge morsen:

ROBOTERGESCHWAETZ