

Dozent

Prof. Dr. Thomas Vetter
Departement
Mathematik und Informatik
Spiegelgasse 1
CH – 4051 Basel

Assistenten

Bernhard Egger
Andreas Forster

Tutoren

Marvin Buff
Sein Coray
Eddie Joseph
Loris Sauter
Linard Schwendener
Florian Spiess

Webseite

[http://informatik.unibas.ch/hs2017/
uebung-erweiterte-grundlagen-der-programmierung/](http://informatik.unibas.ch/hs2017/uebung-erweiterte-grundlagen-der-programmierung/)

Erweiterte Grundlagen der Programmierung (45398-01)**Blatt 2****[10 Punkte]**

Vorbesprechung 2. Okt - 6. Okt

Abgabe 9. Okt - 13. Okt (vor dem Tutorat)

Wir empfehlen Ihnen, im Buch “Sprechen Sie Java” bis und mit Kapitel 4 zu lesen, bevor Sie beginnen die Übungen zu lösen.

Aufgabe 1 - Domino**[2 Punkte]**

Schreiben Sie ein Programm, dass alle möglichen Domino-Spielsteine ausgibt, verwenden Sie dazu zwei geschachtelte Schleifen. Zählen sie jeweils die Anzahl möglicher Steine.

(0|0), (0|1), ..., (1|6), ..., (6|6)

(a) Es dürfen doppelte Steine vorkommen wie z.B. (1|6) und (6|1). **[1 Punkt]**

(b) Es dürfen keine doppelte Steine vorkommen (mit Schleifen lösbar). **[1 Punkt]**

Aufgabe 2 - Zeichnen**[4 Punkte]**

In dieser Aufgabe sollen Sie einfache Muster in ein Fenster zeichnen. Verwenden sie die Sourcen von der Übungswebseite. Die Datei `BasicDrawing.java` im Wurzelverzeichnis können Sie als Vorlage für die Programme in dieser Aufgabe verwenden. Es sollte sich ein Fenster mit einem einsamen roten Pixel auf dem weissen Hintergrund öffnen. Das Programm läuft weiter solange das Fenster nicht geschlossen wird.

(a) Schreiben Sie ein Programm welches eine horizontale und vertikale Linie über das ganze Bild zeichnet. Verwenden Sie dafür eine Schleife. **[$\frac{1}{2}$ Punkt]**

(b) Schreiben Sie ein Programm welches ein gefülltes Rechteck zeichnet. Verwenden Sie dafür zwei geschachtelte Schleifen. **[$\frac{1}{2}$ Punkt]**

(c) Schreiben Sie ein Programm welches ein Schachbrettmuster über das ganze Bild zeichnet. Die Kantenlänge jedes Feldes soll 20 Pixel betragen. Verwenden Sie zwei ineinander geschachtelte Schleifen über die Zeilen und Spalten des Bildes, und eine Bedingung die an jedem Pixel testet, ob dieser Schwarz oder Weiss gezeichnet werden soll.

Hinweis: Verwenden Sie für die Bedingung die Modulo Operation `%`. **[1 Punkt]**

(d) Zeichnen Sie einen gefüllten Kreis um den Nullpunkt des Bildes. Für jeden Punkt auf einem Kreis um den Nullpunkt gilt

$$r^2 = x^2 + y^2 \quad (1)$$

wobei r der Radius des Kreises ist und x, y die Koordinaten eines Punktes auf dem Kreis sind. Überlegen Sie, welche Bedingung im Inneren eines Kreises gilt, und markieren Sie alle Pixel die diese Bedingung erfüllen. [1 Punkt]

(e) Verschieben Sie den Kreis in die Mitte des Bildes. [1 Punkt]

Aufgabe 3 - Iterationsverfahren nach Newton [4 Punkte]

Die Kubikwurzel einer positiven reellen Zahl a lässt sich näherungsweise durch die Iterationsformel bestimmen:

$$x_{n+1} = \frac{1}{3} \left(2x_n + \frac{a}{x_n^2} \right) \quad (2)$$

Diese Formel wird wie folgt angewandt. Man wählt einen beliebigen Startwert x_1 , und berechnet mit der Formel den Wert x_2 , indem $x_n = x_1$ und $x_{n+1} = x_2$ gewählt werden. Das Ergebnis wird dann immer wieder (iterativ) in die Formel hineingesteckt. Die Formel ist so konstruiert, dass die Lösung immer dichter an der Kubikwurzel von a liegt, als der Eingabewert. Die Formel ergibt sich aus dem Iterationsverfahren nach Newton. Um die Nullstelle einer Funktion f zu finden, nutzt man die allgemeine Iterationsvorschrift:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \quad (3)$$

In unserem Fall ist $f(x) = x^3 - a$.

Schreiben Sie ein JAVA-Programm `CubicRoot`, das die Kubikwurzel der Eingabe berechnet. Dabei gilt ein Iterationswert als gut genug, falls er von dem nachfolgendem Iterationswert nicht um mehr als $1e-8$ abweicht. Hinweise:

- Deklarieren Sie die Variablen als Typ `double`.
- Weisen Sie zur Verarbeitung der Eingabe a den Wert `Double.parseDouble(args[0])` zu.
- Starten Sie mit dem Iterationswert 1.
- Nutzen Sie zur Berechnung des Absolutbetrags einer Zahl x die Funktion `Math.abs(x)`.