

Dozenten

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

Assistent

Dr. Marcel Lüthi
Departement
Mathematik und Informatik
Spiegelgasse 1
CH – 4051 Basel

Tutoren / Tutorinnen

Claudia Grundke
Viktor Gsteiger
Simon Dold
Timo Steinebrunner
Alexander Rovner
Nikodem Kernbach
Lukas Stöckli

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

Vorbesprechung 30. September. - 4. Oktober

Abgabe 11. Oktober

Allgemeine Hinweise

- Wir empfehlen Ihnen, dass Sie im Buch “Sprechen Sie Java” bis und mit Kapitel 4 lesen.

Voraussetzung

- Es gelten dieselben Voraussetzungen wie für Übungsblatt 1. Wenn Sie sich betreffend der Umgebung oder der automatisierten Tests noch unsicher sind, lesen Sie bitte nochmals sorgfältig das Infoblatt oder Fragen Sie die Tutoren.
- Die Zip-Datei, die auch dieses Übungsblatt enthält, muss entpackt werden. Es enthält die gesamte Übungsumgebung inklusive der automatisierten Tests. Schreiben Sie ihre Lösungen in die dafür vorgesehenen Dateien, wie in der jeweiligen Übungsaufgabe angegeben.

Empfohlenes Vorgehen

- Wechseln Sie in den Ordner `src/main/java`. Dort finden Sie die Dateien, in welche Sie ihren Java Code schreiben.
- Schreiben Sie ihr Programm, kompilieren Sie dieses mit dem Java Compiler `javac` und führen Sie es mit `java` aus, wie es in der Vorlesung gezeigt wurde.
- Wenn Sie denken, dass alles in Ordnung ist, wechseln Sie zurück ins Übungsverzeichnis `uebung2` und führen da `gradlew test` aus um zu überprüfen ob Ihre Lösung die automatisierten Tests besteht. Überprüfen Sie auch Ihren Codestil mit `gradlew checkstyleMain`. Falls Ihr Code den Vorgaben entspricht, erhalten Sie einen Bonuspunkt.

Abgabe

Ergänzen Sie die Datei `email.txt` mit Ihrer Unibas E-Mail Adresse. Erstellen Sie eine Zip-Datei der gesamten Übungsumgebung (also des Verzeichnisses `uebung2`) und laden Sie dieses auf Courses (<https://courses.cs.unibas.ch/>) hoch.

Aufgabe 1 - Quersumme

[2 Punkte]

Schreiben Sie ein Programm, welches eine Zahl einliest und deren Quersumme ausgibt. Bei negativen Zahlen soll die Quersumme negativ sein.

Schreiben Sie Ihre Lösung in die bereits vorbereitete Datei `CrossSum.java`, die Sie im Verzeichnis `uebung2/src/main/java` finden.

Das Programm sollte sich wie folgt verhalten: Wenn der Benutzer die Zeile

```
java CrossSum 17
```

auf der Konsole eingibt, soll das Programm die Zeile

```
8
```

ausgeben.

Hinweise: Um Parameter welche Java übergeben werden Variablen vom Typ Integer zuzuweisen können Sie folgenden Code benutzen:

```
int a = Integer.parseInt(args[0]);
```

Für die Lösung brauchen Sie die Ganzzahldivision `/` sowie die modulo Operation `%`.

Aufgabe 2 - Iterationsverfahren nach Newton

[4 Punkte]

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

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

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.

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)`.

Schreiben Sie Ihre Lösung in die bereits vorbereitete Datei `CubicRoot.java`, die Sie im Verzeichnis `uebung2/src/main/java` finden.

Anmerkung für Interessierte: Die obige Formel ergibt sich aus dem Iterationsverfahren nach Newton. Um die Nullstelle einer Funktion f zu finden, nutzt man die allgemeine Iterationsvorschrift:

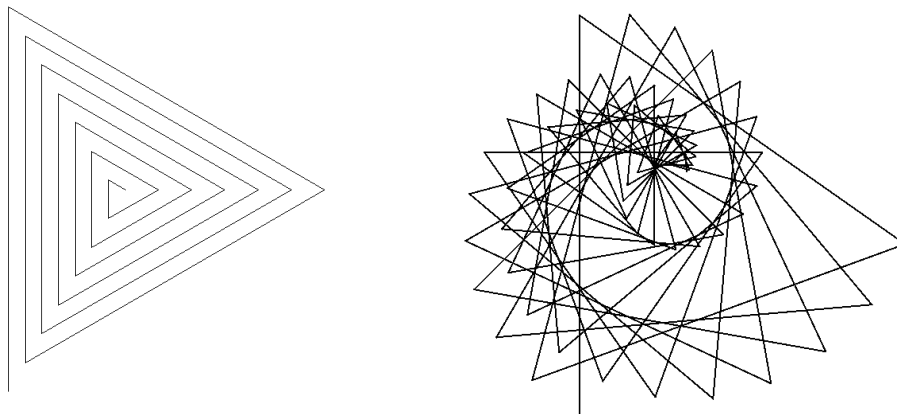
$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \quad (2)$$

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

Aufgabe 3 - Turtlegrafik: Muster reproduzieren

[4 Punkte]

Versuchen Sie folgende zwei Muster mit Turtlegrafik zu reproduzieren: Es ist dabei nicht wichtig, dass Sie genau dieselben Linienabstände oder Winkel wählen. Wichtig ist der allgemeine Aufbau des Musters.



Das Programm sollte sich wie folgt verhalten: Wenn der Benutzer das Programm mit den Argumenten `1 output.png` aufruft, soll das Programm das erste Muster in die Datei `output.png` schreiben. Entsprechend sollen der Aufruf mit den Argumenten `2 output.png` das zweite Muster produzieren.

Schreiben Sie Ihre Lösung in die bereits vorbereitete Datei `TurtlePatterns.java`, die Sie im Verzeichnis `uebung2/src/main/java` finden.

Dieses Programm nutzt wieder die Java Bibliothek (`jturtle`), die Sie beim kompilieren explizit mitangeben müssen. Sie kompilieren die Datei wie folgt:

```
> javac -cp .;jturtle-0.5.jar TurtlePatterns.java (Windows)
> javac -cp .:jturtle-0.5.jar TurtlePatterns.java (Linux und MacOS)
```

Entsprechend ist das Kommando zum Aufruf nun

```
> java -cp .;jturtle-0.5.jar TurtlePatterns 1 output.png (Windows)
> java -cp .:jturtle-0.5.jar TurtlePatterns 1 output.png (Linux und MacOS)
```