

Dozent

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

Assistenten

Marcel Lüthi

Tutoren

Pascal Mafli
Loris Sauter
Linard Schwendener
Clemens Büchner
Florian Spiess
Jonathan Aellen
Lukas Stöckli

Erweiterte Grundlagen der Programmierung (45398-01)**Blatt 4****[8 Punkte]**

Vorbesprechung 15. Okt - 19. Okt
Abgabe 22. Okt - 26. Okt (vor dem Tutorat)

Wir empfehlen Ihnen, dass Sie im Buch “Sprechen Sie Java” bis und mit Kapitel 7, sowie Kapitel 10 und 11 lesen, bevor Sie beginnen die Übungen zu lösen.

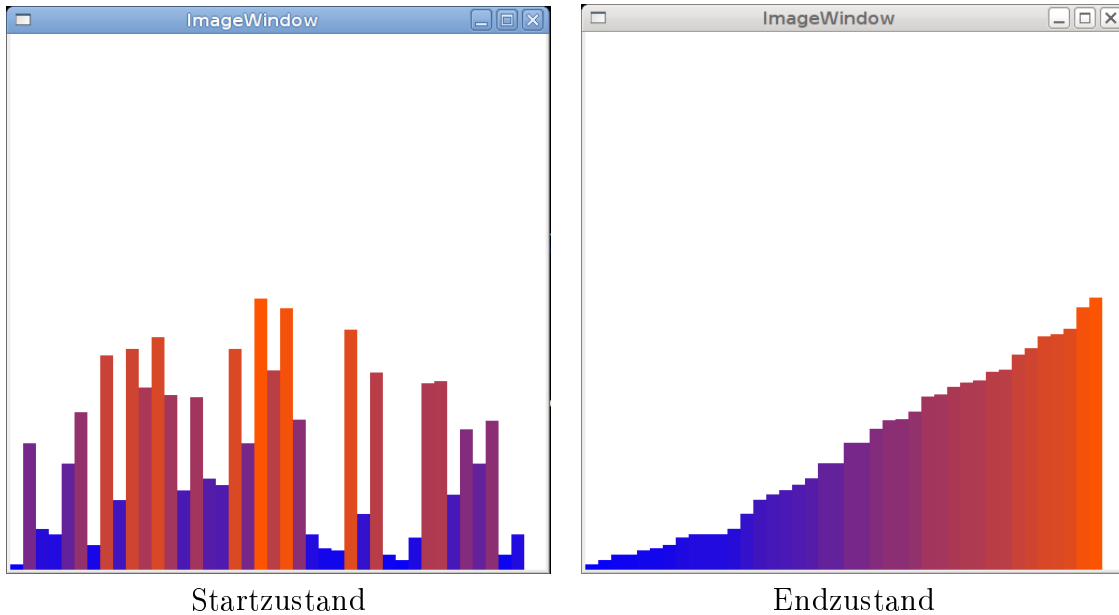
Aufgabe 1 - Bubble Sort-Visualisierung**[4 Punkte]**

Veranschaulichen Sie den Bubblesort-Algorithmus vom vorhergehenden Aufgabenblatt. Haben Sie diese Aufgabe nicht gelöst, können Sie auch die Klasse `HeapSort.java` von der Übungswebseite herunterladen.

Zeichnen Sie dafür nach jedem Vertauschungsschritt das zu sortierende Array als eine Reihe von Balken in das Fenster. Die Höhe der Balken soll den Werten in dem Array entsprechen, und die Farbe der Balken soll ebenfalls vom Wert abhängen. Die Breite des Balkens soll abhängig von der Fensterbreite so gewählt werden, dass alle Balken zusammen einen möglichst grossen Bereich des Fensters ausfüllen.

Es werden folgende Befehle aus dem `ImageWindow` benötigt:

- `w = new ImageWindow(500,500);`
- `w.openWindow();`
- `w.redraw();`
- `w.pause(10);` // zwischen den Animationsschritten
- `w.setPixel(x, y, red , green, blue);`

**Aufgabe 2 - Turtle-Grafik**

[4 Punkte]

In dieser Aufgabe werden Sie eine einfache Turtle-Grafik implementieren. Die Grundidee bei einer Turtle-Grafik besteht darin, dass eine stifttragende ‘Schildkröte’ sich durch einfache Befehle über ein Stück Papier bewegen lässt. Dabei kann der Stift auf dem Papier oder in der Luft sein, so dass der zurückgelegte Weg markiert wird oder nicht. Die Turtle kann sich entweder eine bestimmte Distanz nach vorne bewegen, oder sich um ihre eigene Achse im oder gegen den Uhrzeigersinn drehen. Durch diese einfachen Anweisungen lassen sich komplexe Muster entwerfen.

Verwenden Sie für diese Aufgabe die Klasse `BasicDrawing` vom zweiten Aufgabenblatt als Grundgerüst und die Klasse `ImageWindow` um Pixel zu zeichnen. Benennen Sie jedoch die Klasse `BasicDrawing` um in `BasicTurtle`.

- (a) Erstellen Sie eine neue Klasse `Turtle` und fügen Sie der Klasse Felder für ein `ImageWindow`, die Position als `x`- und `y`-Koordinate, die Laufrichtung der Turtle sowie ein Feld für die Position des Stiftes hinzu. [$\frac{1}{2}$ Punkt]
- (b) Implementieren Sie einen Konstruktor welcher als Argument ein `ImageWindow` entgegen nimmt. Mit dem Argument soll das Klassenfeld initialisiert werden. Weiter soll die Position auf die Mitte des Fensters und die Laufrichtung in Richtung positiver `x`-Achse gesetzt werden. Der Stift soll in der Luft initialisiert werden. [$\frac{1}{2}$ Punkt]
- (c) Schreiben Sie eine Methode welche den Stift in die jeweils andere Position bewegt und eine Methode welche die Turtle je nach Parameterwert um ein Vielfaches von 90 Grad im Uhrzeigersinn oder gegen den Uhrzeigersinn drehen kann. [1 Punkt]
- (d) Schreiben Sie eine Methode mit variabler Parameterliste welche die Turtle nach vorne bewegt. Verwenden Sie je nach Länge der Parameterliste die Werte als Schrittlänge, roter Farbanteil, grüner Farbanteil und blauer Farbanteil. Bekommt die Methode nicht genügend Parameter, verwenden Sie vordefinierte Werte. Beachten Sie dass die Turtle auf jeden Fall den Weg zurücklegt, markiert wird der Weg jedoch nur wenn der Stift auf dem Blatt ist. Verwenden Sie eine `for`-Schleife über die Länge des zurückzulegenden Wegs und `setPixel` um den Weg zu markieren. [1 Punkt]

- (e) Erstellen Sie in der `main`-Methode Ihrer `BasicTurtle`-Klasse zwei Turtles die Sie wie folgt bewegen sollen:
- i) Bewegen Sie die eine Turtle ein Stück weg von der Mitte ohne den Weg zu zeichnen. Zeichnen Sie dann jeweils mit verschiedenen Parametern Striche gefolgt jeweils von einer Drehung um 90 Grad im oder gegen den Uhrzeigersinn. Als Beispiel dient der rechte Linienzug im folgenden Bild. [$\frac{1}{2}$ Punkt]
 - ii) Verwenden Sie ein Integer-Array der Länge vier für die Farbwerte eines Farbkanaals und die Iterator-Form der `for`-Schleife um mit der zweiten Turtle den geschlossenen Linienzug im folgenden Bild zu zeichnen. Alle Bewegungen und Drehungen sollen dabei in der Schleife gemacht werden. Wichtig ist dass die gleichen Abschnitte die selbe Farbe haben und nicht dass Sie die Farben aus dem Bild perfekt nachahmen. [$\frac{1}{2}$ Punkt]

