

**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 3****[10 Punkte]**

Vorbesprechung 7. Oktober. - 11. Oktober

Abgabe 18. Oktober

**Allgemeine Hinweise**

- Wir empfehlen Ihnen, dass Sie im Buch “Sprechen Sie Java” bis und mit Kapitel 7 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 `uebung3` 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 `uebung3`) und laden Sie dieses auf Courses (<https://courses.cs.unibas.ch/>) hoch.

**Aufgabe 1 - Transponierte einer Matrix**

[2 Punkte]

Sie finden im Verzeichnis `src/main/java/` die Datei `Matrix.java`. Implementieren Sie die Methode `transpose(double[][] matrix)` die eine Matrix von beliebiger Dimension nimmt und die Transponierte dieser Matrix zurückgibt. Die übergebene Matrix darf dabei nicht verändert werden.

**Aufgabe 2 - Bubble Sort I**

[3 Punkte]

Mit dem Verfahren *Austauschsortieren* (Bubble Sort) kann man die Elemente eines Arrays sortieren. Dazu wandert man von vorne nach hinten durch das Array und betrachtet jeweils benachbarte Elemente. Wenn diese in der falschen Reihenfolge stehen, vertauscht man sie. Ist man am Ende des Arrays angelangt, beginnt man wieder von vorne - solange bis keine Vertauschungen mehr auftreten.

Sie finden im Verzeichnis `src/main/java` die Datei `BubbleSort.java`. Implementieren Sie die fehlenden Methoden in dieser Klasse.

**Aufgabe 3 - Pixelgrafik**

[5 Punkte]

Sie finden im Verzeichnis `src/main/java/` die Datei `TurtlePixelGraphics.java`. Darin sind die Methoden `drawPixel` und `newLine` implementiert, mit der Sie mit Turtlegrafik einen Pixel an der aktuellen Turtleposition zeichnen können sowie jeweils auf die nächste Zeile springen können.

Implementieren Sie eine Methode `drawImage(boolean[][] image)`, welches ein 2D boolean Array nimmt, und daraus das resultierende Bild zeichnet, indem es obige Funktionen aufruft.

Um die Methoden zu testen, implementieren Sie eine Methode `implicitCircle` welche die impliziten Gleichung für den Kreis

$$x^2 + y^2 - r^2 < 0$$

einen Kreis mit Radius  $r$  implementiert.

Schlussendlich implementieren Sie die Methode `drawCircleImage`, welche ein Bild (ein 2D boolean Array) eines Kreises mit gegebenem Radius erstellt.

*Hinweis: Beachten Sie, dass obige Kreisgleichung einen Kreis mit Radius  $r$  mit Mittelpunkt  $(0,0)$  definiert. Sie können also nicht direkt die Indizes ihres Arrays für die Definition nehmen, sondern müssen diese noch entsprechend anpassen, damit der Kreis in die Mitte des Bildes zu liegen kommt.*

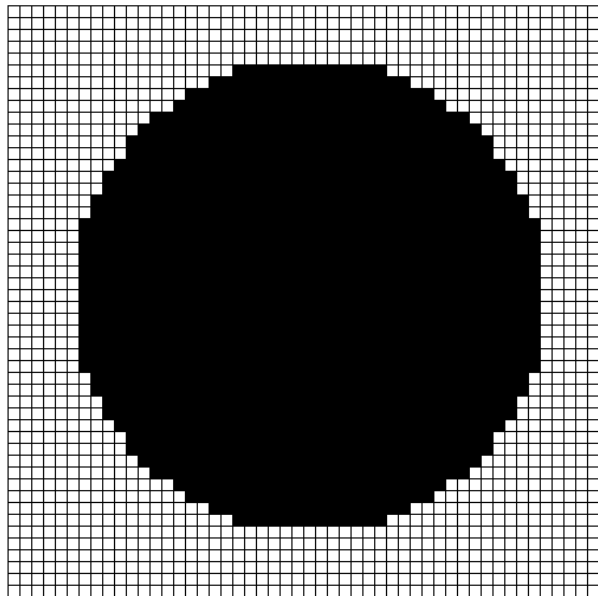
Kompilieren Sie das Programm wie folgt:

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

Wenn Sie das Programm mit

```
> java -cp .;jturtle-0.5.jar TurtlePixelGraphics 20 circleImage.png  
  (Windows)  
> java -cp .:jturtle-0.5.jar TurtlePixelGraphics 20 circleImage.png  
  (Linux und MacOS)]
```

ausführen, sollte folgendes Bild produziert werden.



Freiwillig: Zeichnen Sie Bild eines Herzes, indem Sie die Formel für die Herzkurve nutzen:  
(siehe auch <http://mathworld.wolfram.com/HeartCurve.html>)

$$(x^2 + y^2 - 1)^3 - x^2 y^3 < 0$$