

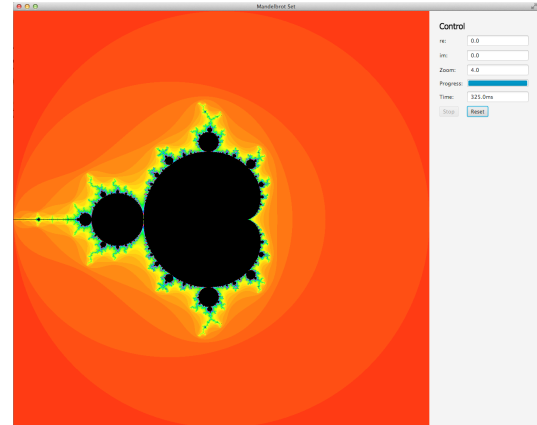
## Übung 1: Mandelbrot Fx Applikation

In dieser Übung modifizieren Sie eine JavaFx Applikation zur Berechnung und Darstellung der Mandelbrot Menge [1].

Die aktuelle Version der Applikation hat zwei Probleme:

1. Das Programm reagiert nicht (hängt) während die Berechnung läuft (Nebenläufigkeit).
2. Die Berechnung nutzt die Verfügbaren Ressourcen nicht aus (Parallelität).

In dieser Übung beheben Sie beide Probleme. Lassen Sie sich nicht abschrecken von der verwendeten Mathematik in diesem Beispiel. Sie müssen die Theorie dahinter nicht verstehen, sondern können einfach nur die schönen Muster bestaunen.



### Aufgaben

1. Importieren Sie das Wochenprojekt 01\_Threads.zip Archiv [2] als Gradle Projekt in Ihren Workspace und führen Sie die Klasse as.JavaFXMandelbrot.java aus. Wenn Sie jetzt ein Fenster analog zum obigen Screenshot sehen, sind Sie bereit für die Aufgabe.

Falls beim Ausführen einer Klasse in Eclipse oder VSCode folgender Fehler auftritt:

```
Error occurred during initialization of boot layer
java.lang.module.FindException: Module javafx.controls not found, required by threads
```

Führen Sie in der Konsole folgendes Kommando aus:

```
WIN> gradlew.bat eclipse      Linux/OSX> ./gradlew eclipse
```

Dabei werden die Metadaten für die IDE so generiert, dass die Javafx Module korrekt gefunden werden.

2. In der Methode JavaFXMandelbrot.drawMandel finden Sie einen klar hervorgehobenen Code Block, der die Berechnung des Bildes und die Zeitmessung beinhaltet. Dieser Block soll neu asynchron in einem eigenen Thread ausgeführt werden. Wenn Sie alles richtig gemacht haben, sollten Sie danach den Fortschritt während einer Berechnung schrittweise verfolgen können.
3. Implementieren Sie die Methode Mandelbrot.computeParallel. Darin soll die Mandelbrotmenge parallel mit mehreren Threads berechnet werden. Sie müssen dazu für jedes Pixel des Bildes (x,y) eine Berechnung durchführen. Orientieren Sie sich an der Methode computeSequential und überlegen Sie sich, wie man das Problem auf mehrere Threads verteilen kann. Hinweis: Nicht nur Brot kann man (z.B. entlang der x-Achse) in Stücke schneiden. Rufen Sie aus der Klasse JavaFXMandelbrot nun ihre neue computeParallel Methode auf statt der lahmen computeSequential.
4. Variieren Sie die Anzahl Threads die am Problem arbeiten und messen Sie die Zeit. Erstellen Sie in Excel o.ä. einen Plot, in dem man die Laufzeit in Abhängigkeit der Anzahl Threads ablesen kann. Wie viel schneller als die sequentielle Lösung werden Sie maximal? Wie erklären Sie die Resultate?
5. [Optional] Implementieren Sie eigene Abbildungen der Anzahl Iterationen auf Farben oder weitere Iterationsformeln, z.B.  $z_{n+1} = z_n^3 + c$

[1] [http://en.wikipedia.org/wiki/Mandelbrot\\_set](http://en.wikipedia.org/wiki/Mandelbrot_set)

[2] 01\_Threads\Lecture\01\_Threads.zip