

## Mandelbrot met Swingworkers

### Leerdoelen

In deze opgave leer je hoe je met behulp van de `SwingWorker`-klasse tijdrovende taken binnen Swing-applicaties kunt uitvoeren. Meer specifiek, na het maken van deze opdracht kun je:

- Klassen specificeren die langdurige taken representeren door de `SwingWorker`-klasse uit te breiden.
  - Ook hier is het weer zaak om uitgaande van een sequentiële versie van het programma een geschikte klasse te kiezen die `SwingWorker` uitbreidt.
  - Na te gaan welke (andere) klasse(n) verantwoordelijk is(zijn) voor het creëren, starten en eventueel beëindigen van zo'n taak.
- Nagaan op welke wijze (tussen)resultaten worden doorgegeven en bepalen hoe die verder worden verwerkt.
- Voortgangsinformatie van taak/taken weergeven in een *voortgangsbalk* (in Java `JProgressBar`).

### Mandelbrot in swing

In de Mandelbrotopdracht hebben jullie gezien hoe je fraaie plaatjes kunt genereren door punten op het scherm op een speciale manier te kleuren. Hiertoe werd aan ieder punt  $(a, b)$  van het platte vlak, waarbij  $a$  en  $b$  reële getallen zijn, een zogenaamd *mandelgetal* toegekend. Dit gebeurde aan de hand van de volgende reeks van punten, startende bij  $(a, b)$ :

$$\begin{aligned}(x_0, y_0) &= (a, b) \\ (x_{n+1}, y_{n+1}) &= (x_n^2 - y_n^2 + a, 2 \times x_n \times y_n + b)\end{aligned}$$

Het mandelgetal is de kleinste waarde van  $n$  waarvoor geldt dat het bijbehorende punt  $(x_n, y_n)$  een afstand van meer dan 2 tot de oorsprong  $(0, 0)$  heeft. Je hoeft deze berekening van het mandelgetal in deze opdracht niet opnieuw te implementeren; die geven we voor. Ook de omrekening van mandelgetallen naar echte kleuren krijg je cadeau. Als je erg trots bent op je eigen implementatie mag je die natuurlijk ook hergebruiken voor deze opdracht.

Zoals je weet, hebben sommige punten het mandelgetal  $\infty$ . Om te voorkomen dat de berekening van het mandelgetal eveneens oneindig is, reken je ten hoogste een bepaald maximum aantal stappen, bijvoorbeeld 200, in bovenstaande reeks uit. Indien de afstand dan nog steeds niet groter dan 2 is nemen we maar aan dat het mandelgetal oneindig is.

Het op deze manier generen van zo'n Mandelbrotplaatje is voor een vaste diepte `MAX_ITERATIONS` (gedefinieerd in de klasse `GridFiller`) al geheel voorgegeven. Wat verder is toegevoegd is een *controller* die op muisklikken reageert: als de gebruiker ergens op het plaatje klikt, dan wordt het aangewezen punt het nieuwe middelpunt en worden lengte en breedte van het oorspronkelijke gebied gehalveerd. Op deze wijze zoom je dus in op het aangeklikte punt. Het hertekenen van het plaatje voor deze nieuwe situatie is ook al voorgegeven.

Wat je als eerste dient toe te voegen is de volgende manier van weergeven: in plaats van het meteen uitrekenen van alle punten op maximale diepte dien je de diepte stapsgewijs te vergoten en het daarbij behorende plaatje te tekenen totdat je de maximale diepte hebt bereikt. Je kunt dit zien in het door Marein (een van onze assistenten) gemaakt filmpje op <https://www.youtube.com/watch?v=YIuQ0iij2CM>.

Mede afhankelijk van hoe je dit precies implementeert, kan zo'n berekening nu best lang duren. Op dit moment wordt het allereerste plaatje door de *main thread* uitgerekend en alle andere plaatjes op de *Event Dispatch Thread*. Zoals jullie weten dien je lange berekeningen op de EDT te vermijden<sup>1</sup>

<sup>1</sup>We hebben het probleem nog wat aangedikt door in de methode `fill` (uit de klasse `GridFiller`) met `sleep` een opzettelijke vertraging in te bouwen.

In deze opdracht lossen we dit probleem op door gebruik te maken van *SwingWorkers*. Bestudeer de slides van het college en het bijbehorende hoofdstuk uit het boek waarin staat uitgelegd hoe je de klasse *SwingWorker* kunt gebruiken.

Je uiteindelijke oplossing dient aan de volgende eisen te voldoen.

1. De swingworker rekent niet het gehele plaatje uit voordat dit getekend wordt. Zodra een deel hiervan (bijvoorbeeld een lijn of kolom van punten) uitgerekend is moet dit worden gepubliceerd waarmee het met de methode `process` op de EDT kan worden getekend.
2. Aan de applicatie dient een progressbar te worden toegevoegd die aangeeft hoe ver het uitrekenen (en weergeven) van het plaatje is gevorderd.
3. Het plaatje moet iteratief worden geconstrueerd, dat wil zeggen, de rekendiepte dient stapsgewijs te vergroot tot het maximum is bereikt.

## Meerdere Swingworkers

Je kunt de berekening nog verder versnellen door niet één maar meerdere Swingworkers te gebruiken die gezamenlijk alle mandelgetallen uitrekenen. Hiervoor kun je het beste het gebied in tweeën of in vieren splitsen en voor ieder deelgebied een aparte singworker te introduceren die verantwoordelijk is voor het uitrekenen van de mandelgetallen. Voeg deze mogelijkheid toe aan je implementatie.

## Optioneel: inzoomen

Het inzoomen gebeurt nu vrij grof. Je kunt dit verfijnen door ook dit stapsgewijs te doen, dus niet in een keer lengte en breedte te halveren, maar in een aantal stappen. Dit alles kun je aanzienlijk versnellen door te bedenken dat je van het nieuwe gebied al een flink aantal punten hebt uitgerekend. Het bijhouden en hergebruiken van (tussen)resultaten kun je overigens ook gebruiken bij het iteratief ophogen van de diepte.

## Opmerking

Het zip-bestand met de voorgegeven klassen dat je op Blackboard kunt vinden is gegenereerd vanuit Netbeans. Als je met Netbeans werkt kun je zonder veel moeite hieruit een project construeren door onder *File* via de optie *Import Project ...* het item *From ZIP ...* te kiezen.

## Inleveren

Voor zondag 7 juni, 11:00 uur, via Blackboard.