

W8 Threads: Ontbinden in factoren en priemgetallen

In deze opdracht gebruiken we parallelle threads om een groot getal in zijn factoren te ontbinden. Indien er geen factoren zijn hebben we te maken met een priemgetal.

Vul de klasse **FactorenRunnable** aan om de gewenste afdruk te verkrijgen. Om de factoren te bepalen kan je gebruik maken van wiskundige algoritmen. Wat we hier doen is een “brute force” methode: we trachten het getal te delen door alle delen tot en met de vierkantswortel uit het getal. In plaats van dit in één thread te doen, splitsen we het werk op.

Om het aantal threads zo goed mogelijk te tunen

- starten we pas een nieuwe thread op als er meer dan 1000 getallen te onderzoeken zijn (
- starten we niet meer threads op als er processoren in onze machine zitten

Op die manier berekenen we het aantal getallen dat elke thread krijgt (blokGrootte). Stel dat de blok grootte 10.000 is, dan krijgt de eerste thread de getallen van 2 (want 1 wordt niet als een factor beschouwd) tot 10.001, de volgende van 10.002 tot 20.001 enz...

Wanneer een deling mogelijk is wordt de gevonden deler (factor) afgedrukt, samen met de naam van de bijbehorende thread.

De Runnable klasse krijgt begin en einde van een blok door en het te testen getal.

Schrijf de run methode. Indien je een deler (factor) gevonden hebt zet dan

- zet je een static boolean op true. Die boolean kan onderzocht worden vanuit de main klasse met de isGevonden() methode.
- druk je deze af
- hoeft je niet verder te zoeken..

```
public class FactorenRunnable implements Runnable{
    private long getal;
    private static boolean gevonden = false;
    private long begin;
    private long einde;

    public FactorenRunnable(long begin, long einde, long
teOnderzoekenGetal) {
        this.begin=begin;
        this.einde=einde;
        getal = teOnderzoekenGetal;
    }

    public static boolean isGevonden() {
        return gevonden;
    }
}
```

Vul ook de klasse ExecuteFactoren aan die de main methode bevat, hierin moet je nog het volgende doen:

- Het nodige aantal threads maken met correcte beginwaarde, eindwaarde en het te zoeken getal (via een for-lus) en ze starten.
- Als alle threads gestart zijn wacht je op de resultaten. Indien er geen delers gevonden werden, druk je er “xxx is een priemgetal!” af.

Bijkomende uitdaging voor wie hierin geslaagd is: wanneer een resultaat gevonden is, wacht dan niet meer op het resultaat van de andere threads.

Enkele voorbeelden van in- en uitvoer (punten tussen de getallen hoeft je niet in te geven). Het getal aan het begin van een regel is LocalTime.now(), want je kan er niet altijd op rekenen dat de console de gegevens in tijdsvolgorde afdrukt.

```
/*
Geef een getal: 123456
Getal= 123456
Hoogst mogelijke factor= 351
#Threads= 1
blok grootte= 352
21:28:42.829: >>> onderzoek van 2 tot 351
21:28:42.829: == 123456 is deelbaar door 2
21:28:42.829: <<< onderzoek van 2 tot 351
*/

/*
Geef een getal: 214.577.422.401
Getal= 214577422401
Hoogst mogelijke factor= 463225
#Threads= 8
blok grootte= 57911
21:27:11.954: >>> onderzoek van 405386 tot 463225
21:27:11.954: >>> onderzoek van 347474 tot 405385
21:27:11.954: >>> onderzoek van 173738 tot 231649
21:27:11.954: >>> onderzoek van 289562 tot 347473
21:27:11.954: >>> onderzoek van 2 tot 57913
21:27:11.954: >>> onderzoek van 231650 tot 289561
21:27:11.970: == 214577422401 is deelbaar door 3
21:27:11.970: <<< onderzoek van 2 tot 57913
21:27:11.954: >>> onderzoek van 115826 tot 173737
21:27:11.954: >>> onderzoek van 57914 tot 115825
21:27:11.970: == 214577422401 is deelbaar door 150049
21:27:11.970: == 214577422401 is deelbaar door 450147
21:27:11.970: <<< onderzoek van 115826 tot 173737
21:27:11.970: <<< onderzoek van 405386 tot 463225
*/

/*
Geef een getal: 14.577.422.401
Getal= 14577422401
Hoogst mogelijke factor= 120736
#Threads= 8
blok grootte= 15100
21:26:41.555: >>> onderzoek van 75507 tot 90607
21:26:41.555: >>> onderzoek van 45305 tot 60405
21:26:41.555: >>> onderzoek van 60406 tot 75506
21:26:41.555: >>> onderzoek van 90608 tot 105708
21:26:41.555: >>> onderzoek van 2 tot 15102
21:26:41.555: >>> onderzoek van 15103 tot 30203
21:26:41.555: >>> onderzoek van 105709 tot 120736
21:26:41.571: <<< onderzoek van 45305 tot 60405
21:26:41.555: >>> onderzoek van 30204 tot 45304
21:26:41.571: <<< onderzoek van 90608 tot 105708
21:26:41.571: <<< onderzoek van 75507 tot 90607
21:26:41.571: <<< onderzoek van 15103 tot 30203
21:26:41.571: <<< onderzoek van 2 tot 15102
21:26:41.571: <<< onderzoek van 60406 tot 75506
21:26:41.571: <<< onderzoek van 105709 tot 120736
21:26:41.571: <<< onderzoek van 30204 tot 45304
14577422401 is een priemgetal!
*/
```

