# Java Advanced – Multithreading

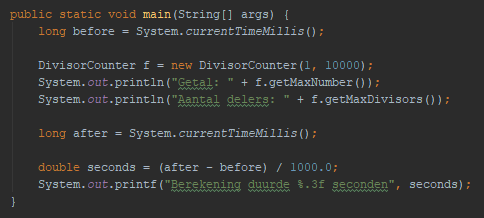
Opdracht 1

We gaan op zoek naar het gehele getal tussen 1 en 10000 met het grootste aantal gehele delers. (rest na deling door de deler = 0) We willen weten welk getal dat is en hoeveel delers het heeft.

Maak een klasse **DivisorCounter**. Aan de constructor kan je een minimum en maximum geven, om de *range* mee te kunnen geven (hier 1 en 10000). Schrijf een methode waarmee je het getal met het hoogste aantal delers binnen deze range kan vinden. Deze methode mag rechtstreeks aangeroepen worden vanuit de constructor.

Het gevonden getal (én het aantal delers) moeten in een member variabele opgeslagen worden.  
Het resultaat zou 7560 of 9240 moeten zijn, beide getallen hebben 64 gehele delers.

Voeg daarna een **main methode** toe die de volgende code uitvoert:



De berekening wordt hier dus opgestart en de duur ervan wordt afgeprint.

Vergroot het bereik eens naar 50000 of 100000 getallen, je ziet dat de duur **exponentieel** groeit.

Pas je code aan zodat je de berekening kan opdelen en elk deel door een **aparte thread** kan laten uitvoeren. Je zal de *DivisorCounter* klasse dus moeten uitbreiden zodat deze *Thread* of *Runnable* implementeert en het aanroepen van de methode naar de *run()* methode verplaatsen.

Nadien schrijf je in de main method de nodige code om verschillende threads met een deel van het bereik op te starten. Wanneer de threads klaar zijn met uitvoeren (gebruik ***join()*** ), doe dan tenslotte de nodige vergelijkingen om te bepalen welk getal het grootste aantal delers heeft van het hele bereik.

Check of je programma nu sneller uitvoert. Met 10 threads kregen wij het volgende resultaat:

