

P2W4 Threads: Opdracht Streams

Om te beginnen moet je de klasse **NumberGenerator** aanvullen, verplicht gebruik van een **Stream** (zie Javadoc van de interface **Stream<T>** of de slides van week 7 – P2W1) .

De klasse **NumberGenerator**

```
public class NumberGenerator {
    private List<Integer> numbers =
        // Werk hier met een Stream

    private List<Integer> bigNumbers =
        // Werk hier met een Stream

    public List<Integer> getNumbers() {
        // Zorg er voor dat de inhoud van numbers niet gewijzigd kan worden
    }

    public List<Integer> getBigNumbers() {
        // Zorg er voor dat de inhoud van bigNumbers niet kan wijzigen
    }

    public void controleAfdruk() {
        numbers.forEach(n -> System.out.print(n + " "));
        System.out.println();
        bigNumbers.forEach(n -> System.out.print(n + " "));
        System.out.println();
    }
}
```

Bij het uitvoeren van de methode **controleAfdruk** (vanuit de klasse **RunStreams**) moet je het volgende bekomen:

```
/*
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
2000 2101 2202 2303 2404 2505 2606 2707 2808 2909 3010 3111 3212 3313 3414 3515
3616 3717 3818 3919
*/
```

De klasse RunStreams

```
public class RunStreams {
    private static final Random generator = new Random();

    private static Runnable getRunnable(List<Integer> list) {
        return () -> {
            // Vul hier de lambda aan

            // 1. Maak een stream op basis van de parameter list
            // 2. Filter de oneven getallen eruit
            // 3. Sorteer de overgebleven getallen in omgekeerde volgorde
            // 4. Zet de stream om naar een list en ken het resultaat toe
            //    aan een nieuwe List<Integer> met de naam myList

            // Draai ten slotte een lus waarbij je elk getal op één regel
            // afdrukt. Zorg bij het afdrukken na elk getal voor een
            // willekeurige vertraging van 0 tot 1000ms.
        };
    }

    public static void main(String[] args) {
        NumberGenerator generator = new NumberGenerator();
        List<Integer> numbers = generator.getNumbers();
        List<Integer> bigNumbers = generator.getBigNumbers();
        // generator.controleAfdruk();

        Runnable numbersRunnable = getRunnable(numbers);
        Runnable bigNumbersRunnable = getRunnable(bigNumbers);

        // Maak hier de 2 threads aan en start ze.
    }
}
```

Mogelijke uitvoer (5 runs):

```
20 3818 3616 3414 18 3212 3010 16 2808 14 12 2606 10 2404 2202 8 2000 6 4 2
3818 20 18 3616 16 3414 3212 14 3010 12 2808 10 2606 8 2404 2202 6 4 2000 2
3818 20 3616 18 3414 3212 16 3010 2808 2606 14 12 2404 10 2202 8 6 4 2000 2
20 3818 3616 18 3414 16 3212 14 3010 12 2808 2606 2404 2202 10 8 2000 6 4 2
20 3818 3616 18 3414 16 3212 3010 14 2808 2606 2404 12 2202 10 2000 8 6 4 2
```