TINPRO04-4: Java Multithreading  
Oefening Les 1: Basic Thread controls

Stap 1: sleep()

In de aangeleverde App.java staat een class Foo, die bij een call naar Foo.loop() een counter van 0 naar 5 laat tellen en bij elke iteratie de counter op het scherm print.

* Zorg dat het programma bij elke iteratie van de loop 1 seconde slaapt. Voeg hiervoor een regel toe aan de loop. Zorg dat je alle bijkomende Exceptions netjes afhandelt

Test dit door een call naar Foo.loop() toe te voegen aan je main(). Je code zou een keer van 0 tot 5 moeten tellen, met telkens een seconde ertussen.

Stap 2: Thread / Runnable class

Deze taak willen we *parallelliseren*, oftewel in meerdere “Threads of execution” tegelijkertijd naast elkaar laten draaien.

* Begin met het omschrijven van class Foo, zodat je er een Thread object van kan maken. Let hierbij goed op [interface Runnable](https://docs.oracle.com/en/java/javase/21/docs/api/java.base/java/lang/Runnable.html): wat moet je overriden van deze interface in class Foo?

Test je code door in main() twee aparte Foo Threads aan te maken, en ze direct achter elkaar op te starten. Let hierbij goed op de method die je gebruikt om ze op te starten!

Als het goed is zouden de twee Threads nu tegelijk van 0 tot 5 moeten tellen.  
  
Merk je nog iets raars aan het gedrag van je programma? Probeer uit te zoeken hoe dat komt.

Stap 3: Pushing the limit

Maak een array van Foo en vul deze met threads met behulp van een loop. Bijv.:

Thread threads[] = new Foo[1000];

Start alle Threads in een keer door met een loop door de array heen te lopen.

Hoeveel Threads kun je starten voordat je programma raar begint te doen? Vergelijk dit met je medestudenten. Probeer redenen te bedenken waar dit aan zou kunnen liggen.