Exercise with executor service:

Preparation:

Go and download the demo and excercise files for today:

https://github.com/sofusalbertsen/Thread-fut-cal

Look at the file SequentialPinger.java

This program takes a list of URL's and makes a HTTP request to the given URL in order to see if the site is up or not.

But the program has no multi threaded part, and is therefore running all the checks sequential. This is a huge bottleneck when the list of websites grows.

Your job is to wrap the functionality into a ExecutorService, and therefore making the checks work in parallel.

Tasks:

- Make an ExcecutorService to handle the tasks.
- Make a task that takes a URL, and returns a response just like getStatus() method
- Print out the statuses afterwards

Remember to shut down your ExcecutorService after use, otherwise your program will not terminate.

Hints are provided on the second page:

Hints

This is a step-by-step guide on how to solve the problem. Try **NOT** to use this all the way. If you get stuck, look at the hints, but continue on your own afterwards.

- You need to make a ExecutorService in order for you to run the tasks. Take the FixedThreadPool() variant, with 10 threads (you can always change that afterwards).
- Just like the CountPrimes class, you need a list of futures of the type String to return your response from the callable.
- Make a loop for every URL, exctract the URL in a final String object
- Make a Callable<String> anonymous inner class
    - In the call() method, copy all the code from getStatus method, and change the url variable with your final String variable created earlier.
- Submit the task to the executor, and add the returned future to the list of futures.
- Loop through the list of futures and print out the status of each site. Remember that the response should include the URL, and not only the status.