Java & Haskell

Resources: for API documentation and Concurrency Tutorial. Cut and paste from these is allowed.

https://docs.oracle.com/javase/8/docs/api/
https://docs.oracle.com/javase/tutorial/essential/concurrency/

1. Look at the available methods in the `Thread` class. Is there a method that will let a thread sleep? How do you call it? [Beware of the exception.]

2. Create a Java class that derives from the `Thread` class. The `run` method of the class should write something to the screen, sleep for a little while, and then write something to the screen again before finally returning.

3. Same as previous but as a class that implements the `Runnable` interface.

4. Write a Java program that creates one thread from each of the classes above and then starts them both. The program should wait until both threads have finished before ending.

5. Create a `Counter` class using either of the above alternatives that has a private (static) tally, a public `getValue` method that returns the current tally, and whose `run` method increments the tally 10^7 times.

6. Run two `Counter` threads in parallel. What is the tally after both threads have terminated?

7. Repeat 2 in Haskell. That is, create and run a Haskell thread that prints, sleeps and prints before exiting. You will need to import `Control.Concurrent` and for sleeping you need `threadDelay`.