Production Quality Software
Professor Schidlowsky

Problem Set 2

Objectives:

- **Implement advice from Effective Java, Chapters 4, 5, and 10.**
- **Practice thinking about concurrency and thread safety.**

Detailed Requirements

For this assignment you will implement a thread-safe stopwatch library. Your library implementation will provide stopwatch objects for timing tasks. The stopwatch objects support the typical operations of a physical stopwatch: start, stop, restart, and the recording of laps (times intervals). Also, a stopwatch can be asked for a list of all the lap times that have been recorded using that stopwatch (if the stopwatch has just been started and stopped once, then the list is of size one and contains the elapsed time).

Both the stopwatch objects and the stopwatch factory need to be thread-safe. That means they might be shared by a number of different threads. You've been provided with skeleton code in the form of two classes

- edu.nyu.pqs.stopwatch.api.IStopwatch - The interface for your Stopwatch implementation class.
- edu.nyu.pqs.stopwatch.impl.StopwatchFactory - The factory class used to create Stopwatch objects.

You've also been provided with a small demo application that uses some of the stopwatch functionality (this application does not prove that your code is threadsafe, it's just a small app that calls some stopwatch methods):

- edu.nyu.pqs.stopwatch.demo.SlowThinker

Do not modify IStopwatch.

Do not modify the *method signatures* defined in StopwatchFactory. Your job is to complete the implementation of the StopwatchFactory class and to create at least one new class that implements the IStopwatch interface.

To get started check the problem set out of your repositories. Please follow these instructions carefully. You will lose points if you modify the IStopwatch class or if you change the method signatures of StopwatchFactory. Also, make sure to incorporate the lessons from "Effective Java." For example, the Stopwatch class should have correct implementations of equals(),

hashCode(), and toString(). Feel free to use the classes in java.util.concurrent where you find them helpful.

Code Review

You have one code review in your project. Please review the code carefully and put your comments in the provided file.

As always, please send any questions to the class mailing list.