PF3: Assignment 4

Due on December 19, 2018 at 8:30am

Prof. Walter Binder

A. Romanelli

Problem 1

Please refer to the attached code for the solution to this problem.

Problem 2

Please refer to the attached code for the solution to this problem.

Problem 3

- Monitor A and Monitor B:
  They can run in parallel because Monitor A is synchronised on the class `Monitors`, whereas the second is on the class `InnerClass`
- Monitor A and Monitor C:
  They will run sequentially because both monitors will require the lock on the same class `Monitors`.
- Monitor A and Monitor D:
  They will be able to run in parallel because they'll require two distinct locks, Monitor A will require the lock on the class `Monitors` whereas the Monitor D will require the lock on the object instantiated by that class, which has a different lock.
- Monitor A and Monitor E:
  In this case it will depend on the parameter `o` passed to the method of Monitor E. They won't be able to run in parallel if the Object passed to the method is `this.getClass()`, that is the Object is the class from which the invoker was instantiated. This means that both Monitor A and E will require the same lock, whereas in all the other cases they will be able to run in parallel.
- Monitor D and Monitor E:
  The same as before applies, but rather than passing the Class object `Monitors`, it will run sequentially when the parameter is given to be `this`, which will cause both monitors to require the same lock. In the rest of the cases it will be possible that both monitors run in parallel.

Problem 4

1. The reason why this program does not scale well is because each connection is assigned a specific thread, which, in case of numerous concurrent connections will spawn will spawn equally as many threads, which will put both CPU and the memory under heavy load and eventually the program will run out of memory if the garbage collection isn't able of keeping up with the spawning of new threads.
2. Check the attached code for the solution to this part of the problem.