Lab 5: Synchronisation and Deadlock

February 26, 2015

The code provided implements a simple client-server banking application. This version of the code implements just one operation – transferring funds between two bank accounts. Further details are provided in Javadoc comments within the code.

1. Study the `Account` class carefully and answer the following:
   (a) Describe a data race present in this class.
   (b) Describe a race condition that is present in this class.
   (c) Why is the constructor in a **`synchronized`** block?
   (d) Which methods require synchronisation to avoid these problems?
2. (a) Fix the problems identified in Q1 by adding proper synchronisation.
   (b) Check whether your the synchronisation you added can cause starvation. Explain why.
   (c) Check whether your the synchronisation you added can cause deadlock. If so, describe the deadlock scenario and how it can be avoided.