Sistemi Operativi e Reti – Estratto dalla prova di Sistemi Operativi del 19 Febbraio 2014

ESERCIZIO Conto Bancario

Si deve implementare una struttura dati per la gestione di transazioni tra più conti bancari, dove le transazioni possono essere *simultanee*.

A tal proposito un **ContoBancario** è composto da:

1. Una variabile *Saldo,* indicante un importo in Euro;
2. Un elenco, in ordine temporale, delle ultime 50 transazioni effettuate sul conto. Una Transazione è composta da
    a. Un ContoBancario sorgente;
    b. Un ContoBancario di destinazione;
    c. Un Valore in Euro;

Ad esempio, la *Transazione* <A,B,10> indica il trasferimento di 10 Euro dal conto A al conto B.

La struttura dati **Banca** deve contenere al suo interno:

1. Una collezione di *ContiBancari (*si supponga di dover gestire circa un milione di conti bancari);
2. Il metodo *int getSaldo(C)*: restituisce l'attuale disponibilità sul conto C;
3. Il metodo *bool trasferisci(A,B,N):* trasferisce N Euro dal conto A al conto B (dove A e B sono conti presenti nella collezione di Conti Bancari della Banca. Il metodo fallisce restituendo *false* se la disponibilità sul conto A è insufficiente ad effettuare l'operazione. Se la transazione viene effettuata, le disponibilità sui conti A e B vengono opportunamente aggiornate, nonché l'elenco delle ultime operazioni registrate in A e B.

I metodi *getSaldo* e *trasferisci* devono essere Thread-safe. Si assuma di prevedere un volume di accessi alla struttura dati di circa 1000 thread in contemporanea.

In ordine di priorità, il codice deve essere implementato:

1. garantendo **la mutua esclusione** nell'accesso ai dati condivisi **solo *ove necessario****;*
2. garantendo l'assenza di situazioni di stallo permanente (deadlock);
3. garantendo il massimo grado di parallelismo ed efficienza; Si assuma di prevedere un volume di accessi alla struttura dati di circa 1000 thread in contemporanea.
4. garantendo l'assenza di situazioni di stallo temporaneo (starvation).

È parte integrante *di questo esercizio completare le specifiche date nei punti non esplicitamente definiti, introducendo tutte le strutture dati che si ritengano necessarie, e risolvendo eventuali ambiguità. Non è consentito modificare il prototipo dei metodi se questo è stato fornito.*

È consentito usare qualsiasi funzione di libreria di Java 6.

Non è esplicitamente richiesto di scrivere un main() o di implementare esplicitamente dei thread di prova, anche se lo si suggerisce per testare il proprio codice prima della consegna (per evitare di bloccare il PC ***non*** *lanciate realmente 1000 thread).*