

L'obiettivo di questa serie d'esercizi è di consolidare l'utilizzo dei read-write locks e di imparare ad utilizzare le operazioni di compare-and-set delle variabili atomiche. Inoltre, permette di capire i meccanismi relativi alla visibilità della memoria e conoscere le tecniche di programmazione concorrente ad oggetti legate agli oggetti immutabile.

Esercizio 1

Copiate il codice sorgente 'S4Esercizio1.java' ed analizzatelo nel dettaglio. Provate a sostituire l'utilizzo dei locks impliciti presenti nel codice, inizialmente con locks espliciti, poi con read-write locks. Confrontate i tempi d'esecuzione delle tre versioni e motivate le differenze riscontrate.

Esercizio 2

Copiate il codice sorgente 'S4Esercizio2.java' ed analizzatelo nel dettaglio. Il programma è una possibile soluzione con le variabili atomiche dell'esercizio 2 (Sensori) della serie precedente. Modificate la logica della classe sensore in modo da sfruttare l'operazione di compare-and-set delle variabili atomiche.

Esercizio 3

Copiate il codice sorgente 'S4Esercizio3.java' ed analizzatelo nel dettaglio. Il codice simula i traslochi di un uomo d'affari che ad ogni spostamento aggiorna i propri numeri di telefono e di alcuni clienti che vogliono restare aggiornati sui suoi recapiti telefonici. Cercate di capire di che problemi soffre l'implementazione e proponete delle soluzioni. Infine, implementate una versione che ritenete efficace, motivando la scelta. Riuscite ad implementare una versione che non faccia del tutto uso di lock (impliciti o espliciti)?

Esercizio 4

Copiate il codice sorgente 'S4Esercizio4.java' ed analizzatelo nel dettaglio. Il programma cerca di simulare un apparecchio GPS che emette una nuova posizione ogni [1; 5] ms mentre il main thread tiene traccia degli spostamenti, calcolando la distanza tra il punto attuale e quello precedente.

Descrivete i problemi che riscontrate e implementate una soluzione che sfrutti gli oggetti immutabile.

Esercizio 5

Scrivete un programma che simuli l'attività di una società attiva nel settore del cambio di valuta. La società è costituita da una sede centrale e da 10 sportelli. Il programma deve implementare la classe 'TassiDiCambio', che contenga una matrice di valori double rappresentante i tassi di cambio tra le valute CHF, EUR, USD, GBP e JPY. La sede centrale (il main thread dell'applicazione) si deve occupare di calcolare ripetutamente i nuovi tassi d'interesse (valore random tra 0.5 e 1.5) e di trasmetterli ai propri sportelli ogni 100 ms. Ogni sportello deve essere rappresentato da un thread indipendente, che simuli, ripetutamente, l'operazione di cambio tra 2 diverse valute scelte a caso su un quantitativo di denaro da cambiare casuale tra 50 e 500. Tra ogni operazione di cambio deve passare un tempo tra 1 e 4 ms.

Implementate il programma utilizzando oggetti immutabile per la classe TassiDiCambio e reentrant locks per la condivisione degli oggetti immutabile.