Università di Ferrara – Informatica
Corso di Sistemi Operativi e Laboratorio
Prova di Java
14 Luglio 2020

10 CFU, durata 40'

Si realizzi un programma Java multi-thread per monitorare il mercato azionario. Il thread *Main* crea una struttura dati condivisa chiamata *DatiAzionari* che espone i seguenti metodi: *getIndice()* e *setIndice()* per recuperare e modificare il nome dell'indice da monitorare, *getValore()* e *setValore()* per scrivere e recuperare l'attuale valore dell'indice.

Il thread *Main* richiede all'utente (via terminale) il nome dell'indice azionario da monitorare e utilizza un opportuno metodo di *DatiAzionari* per indicare l'indice da monitorare. Dopodiché il thread *Main* crea il thread *RecuperaDati* che produce ogni 5 secondi il valore corrente dell'indice dato (valore intero random tra 20 e 80). A tal fine prima di generare un nuovo valore il thread *RecuperDati* deve verificare se l'indice da monitorare è uguale o diverso dal precedente; se è diverso scrive un massaggio di warning sullo standard output. In ogni caso il valore generato deve essere inserito opportunamente nell'oggetto *DatiAzionari*.

Il thread *RecuperaDati* deve estendere la classe Thread. La soluzione deve essere thread-safe.

Il thread *Main* legge il valore corrente su *DatiAzionari* ogni 7 secondi e termina l'esecuzione se il valore corrente è superiore a 70 o inferiore a 30. Prima della terminazione il thread *Main* deve terminare il thread *RecuperaDati* invocandone un opportuno metodo *terminaThread()* e attenderne l'effettiva terminazione. Attenzione, il thread *RecuperaDati* deve terminare immediatamente, senza attendere la terminazione dei 5 secondi tra un ciclo e l'altro.

12 CFU, durata 1h:20'

Si realizzi un programma Java multi-thread per monitorare il mercato azionario. Il thread *Main* crea una struttura dati condivisa chiamata *DatiAzionari* che espone i seguenti metodi: *getIndice()* e *setIndice()* per recuperare e modificare il nome dell'indice da monitorare, *getValore()* e *setValore()* per scrivere e recuperare l'attuale valore dell'indice.

Il thread *Main* richiede all'utente (via terminale) il nome dell'indice azionario da monitorare e utilizza un opportuno metodo di *DatiAzionari* per indicare l'indice da monitorare. Dopodiché il thread Main crea il thread *RecuperaDati* che produce ogni 5 secondi il valore corrente dell'indice

dato (valore intero random tra 20 e 80). A tal fine prima di generare un nuovo valore il thread *RecuperDati* deve verificare se l'indice da monitorare è uguale o diverso dal precedente; se è diverso scrive un massaggio di warning sullo standard output. In ogni caso il valore generato deve essere inserito opportunamente nell'oggetto *DatiAzionari*.

Il thread *RecuperaDati* deve estendere la classe Thread. La soluzione deve essere thread-safe.

Il thread *Main* legge il valore corrente su *DatiAzionari* ogni 7 secondi. Ogni volta che il valore corrente è superiore a 70 o inferiore a 30 invia un messaggio al thread *AvvisaConsumatori* (che implementa l'interfaccia *Runnable*). In particolare i thread *Main* e *AvvisaConsumatori* comunicano tramite una PipedStream di tipo Object, dove il messaggio è composto dalla classe Warning contenente sia il valore appena letto che una stringa "rialzo eccessivo" o "ribasso eccessivo" nel caso in cui il valore corrente sia superiore a 70 o inferiore a 30 rispettivamente. Il thread *AvvisaConsumatori* riceve i vari messaggi inviati dal thread *Main* e ne stampa sullo standard output la stringa ricevuta; tale thread termina quando riceve un oggetto Warning con messaggio "fine".

Il thread *Main* dopo 15 valori azionari letti termina il thread *RecuperaDati* invocandone un opportuno metodo *terminaThread()*, invia al thread *AvvisaConsumatori* un oggetto Warning con stringa "fine", attende la terminazione dei thread e infine termina. Attenzione, il thread *RecuperaDati* deve terminare immediatamente, senza attendere la terminazione dei 5 secondi tra un ciclo e l'altro.