Università di Ferrara
Laurea Triennale in Informatica
A.A. 2021-2022
Sistemi Operativi e Laboratorio

Lab-12. I Thread in Java
Prova d'esame

Prof. Carlo Giannelli

http://www.unife.it/scienze/informatica/insegnamenti/sistemi-operativi-laboratorio
http://docente.unife.it/carlo.giannelli
https://ds.unife.it/people/carlo.giannelli

Esercizio 7: 1/2

Si realizzi un programma Java multithread che simuli il comportamento di una catena di produzione di un'industria 4.0. A tal fine si implementi un thread «Macchina» e una classe «Produzione». Si noti che nella catena di produzione operano più macchine, i cui dati sulla produzione vengono memorizzati in una struttura dati Produzione.

Il thread «**Macchina**» simula il comportamento di una macchina e produce un nuovo prodotto in un intervallo compreso fra 500 e 600 ms. Una volta realizzato il prodotto il thread Macchina va a incrementare il valore dei prodotti in **Produzione** e poi incomincia a realizzare il prodotto successivo.

Si supponga che l'aggiornamento del valore dei prodotti nell'oggetto Produzione sia un'operazione che richiede un tempo variabile, dovuto ai tempi di latenza di un fittizio database legacy. Si implementi tale evento in Produzione simulando un interrupt hardware di 200 ms nel 25% delle operazioni di aggiornamento.

Esercizio 7: 2/2

Il **main-thread** deve occuparsi di creare 10 thread di tipo Macchina e di mandarli in esecuzione. Ogni thread di tipo macchina deve creare 35 prodotti. Infine, una volta mandati in esecuzione i thread macchina, il thread main ne attende la terminazione.

Alla fine dell'esecuzione si verifichi che la classe Produzione registri un risultato corretto. Nel caso in cui il risultato non sia corretto si vada a modificare il programma in modo da garantirne un funzionamento thread-safe.

Esercizio 7: traccia

- Definire la classe **Produzione**
  - con un metodo **incremento()** per consentire di aggiornare il contatore dei prodotti. Il metodo incremento() deve simulare nel 25% dei casi un interrupt hardware tramite una sleep di 200 ms;
  - con un metodo **stampaProduzione()** che visualizzi a video il numero di pezzi prodotti.
- Definire un thread **Macchina** che venga inizializzata passandogli l'oggetto condiviso Produzione. Il thread macchina deve creare 35 prodotti e per ogni prodotto creato deve aggiornare il valore di Produzione.
- Definire un metodo **main** all'interno del quale:
  - creare un oggetto di tipo Produzione;
  - creare e attivare 10 thread di tipo Macchina;
  - effettuare il join dei thread creati per aspettarne la terminazione;
  - utilizzare il metodo stampaProduzione() sull'istanza di Produzione.

Esercizio 8: 1/2

Si realizzi un programma Java multithread che simuli il comportamento di due macchine industriali che con un'interazione ordinata realizzano un prodotto finito in due fasi. A tal fine si implementino le due macchine come un thread «MacchinaA» e un thread MacchinaB.

Il primo thread «**MacchinaA**» si occupa della prima fase della lavorazione, trasformando (in un ciclo infinito) le materie prime in un primo semilavorato. Il secondo thread «**MacchinaB**» invece lavora sui semilavorati (ricevuti da MacchinaA) per produrre prodotti finiti. Ogni fase della lavorazione (semilavorato e prodotto finito) richiede 200 ms.

Si implementi l'interazione tra le due fasi utilizzando i costrutti Piped per la comunicazione fra thread visti a lezione. Per semplicità si ipotizzi che il semilavorato sia una stringa qualunque che MacchinaA deve passare al thread MacchinaB.

Esercizio 8: 2/2

Il **main-thread** deve occuparsi di creare i due thread e mandarli in esecuzione. Una volta mandati in esecuzione si deve sospendere per 10 secondi e poi terminare l'attività di produzione e attendere l'effettiva terminazione dei thread MacchinaA e MacchinaB.

Si realizzi la comunicazione fra i due thread utilizzando le tre modalità viste a lezione: stream, buffered e object.

Si realizzi la terminazione dei thread MacchinaA e MacchinaB con e senza l'uso del metodo interrupt().

Esercizio 8: traccia

- Definire il thread **MacchinaA** che realizza in un ciclo infinito la produzione di un semilavorato in un tempo di 200 ms. Una volta terminato il semilavorato corrente, MacchinaA scrive una stringa su pipe per comunicare a MacchinaB la produzione del semilavorato. MacchinaA deve esporre il metodo stop() per supportare la sua corretta terminazione.
- Definire il thread **MacchinaB** che realizza in un ciclo infinito la produzione dei lavorati finiti a partire dai semilavorati ricevuti da MacchinaA in un tempo di 200 ms. MacchinaB riceve i semilavorati da MacchinaA attraverso la pipe. MacchinaB deve esporre il metodo stop per supportare la sua corretta terminazione.
- Definire un metodo **main** all'interno del quale
  - creare la pipe;
  - creare/attivare i due thread MacchinaA e MacchinaB;
  - attendere 10 s e poi terminare la produzione (ricordandosi di attendere l'effettiva terminazione dei thread).

Esercizio 9: 1/2

Sulla base degli esercizi precedenti si realizzi la seguente applicazione Java thread-safe. In questa versione più sofisticata però i thread **MacchinaA** e **MacchinaB** comunicano attraverso un oggetto condiviso e thread-safe chiamato **ControlloProduzione**.

Il thread MacchinaA realizza un semilavorato in un intervallo di tempo variabile tra 400 ms e 500 ms. Una volta realizzato tale semilavorato, MacchinaA inserisce tale informazione all'interno di ControlloProduzione andando ad incrementare il numero dei semilavorati attualmente disponibili.

Il thread MacchinaB trasforma un semilavorato in un prodotto finito. A tal fine MacchinaB controlla che all'interno di ControlloProduzione sia disponibile almeno un semilavorato. Se tale condizione non è verificata MacchinaB si sospende (tramite sleep) per 2 s. Una volta verificata la presenza di un semilavorato, MacchinaB decrementa il valore dei semilavorati in ControlloProduzione e procede con la creazione di un prodotto finito in un intervallo di tempo tra 100 ms e 150 ms.

Esercizio 9: 2/2

Terminata la produzione, MacchinaB comunica l'evento al **thread-main** utilizzando una pipe. Anche in questo caso si supponga che tale evento venga comunicato utilizzando una stringa arbitraria. Inoltre MacchinaB aggiorna la quantità di lavorati finiti disponibili in ControlloProduzione.

Il thread-main deve creare l'oggetto condiviso ControllaProduzione, 2 thread MacchinaA e 1 thread MacchinaB.

Infine il thread-main dopo essere stato notificato della avvenuta produzione di 15 prodotti finiti termina i thread MacchinaA e MacchinaB e stampa a video il residuo dei semilavorati presenti.

Si verifichi che tali risultati siano corretti.

Esercizio 9: traccia (1/2)

- Definire una classe **ControllaProduzione** che metta a disposizione i metodi per incrementare e decrementate il numero di semilavorati e prodotti finiti in produzione.

- Definire il thread **MacchinaA** che realizza in un ciclo infinito la produzione di un semilavorato in un tempo di 400-500 ms. Una volta terminato il semilavorato corrente, MacchinaA aggiorna il valore dei semilavorati in ControllaProduzione. MacchinaA deve esporre il metodo stop() per consentirne la corretta terminazione.

- Definire il thread **MacchinaB** che realizza in un ciclo infinito la produzione dei lavorati finiti a partire dai semilavorati ricevuti da MacchinaA in un tempo di 100-150 ms. MacchinaB controlla la presenza in ControllaProduzione di almeno un semilavorato. Se tale condizione è verificata procede alla produzione, altrimenti si sospende (con una sleep) per 2 s.

Esercizio 9: traccia (2/2)

- Terminata la produzione, MacchinaB aggiorna il valore dei prodotti finiti in ControllaProduzione e poi **comunica tramite pipe tale evento al thread-main**. MacchinaB deve esporre il metodo stop() per consentirne la sua corretta terminazione.

- Il **thread-main** deve creare una istanza di ControllaProduzione. Poi deve creare due thread di tipo MacchinaA e un thread di tipo MacchinaB. Il thread-main dopo aver ricevuto la notifica dell'avvenuta produzione di 15 prodotti finiti richiede la terminazione dei thread creati (ricordandosi di attendere l'effettiva terminazione dei thread) e stampa a video il numero di semilavorati e prodotti finiti presenti in ControllaProduzione.