

Università di Ferrara
Corso di Sistemi Operativi e Laboratorio
Prova di Java

22 Novembre 2020 – durata 45' per 10 CFU, 1h:30 per 12 CFU

10 CFU

Si realizzi un programma Java multithread che simuli il sistema di gestione della qualità di un'azienda manifatturiera. Per ogni oggetto prodotto una macchina industriale informa un programma di gestione della qualità se tale oggetto è difettoso o meno.

A tal fine il thread *Main* deve creare due thread: *Machine* e *Quality*. *Machine* ogni 750 ms incrementa di uno il numero di prodotti creati (intero), produce un valore casuale tra -1000 e +1000 (float) e invia tali informazioni al thread *Quality*. In caso di float negativo il thread *Quality* deve stampare sullo standard output il messaggio "Errore " e poi il valore intero ricevuto, in caso di float positivo deve scrivere "OK". Per realizzare la comunicazione fra i thread *Machine* e *Quality* si utilizzi il costrutto PipedStream basato sugli oggetti serializzabili visto a lezione. Entrambi i thread devono estendere la classe Thread.

Il thread *Main* dopo aver creato e avviato i thread *Machine* e *Quality* si sospende per 15 secondi, poi termina i due thread, ne attende la loro terminazione e infine termina.

12 CFU

Si realizzi un programma Java multithread che simuli il sistema di gestione della qualità di un'azienda manifatturiera. Per ogni oggetto prodotto una macchina industriale informa un programma di gestione della qualità se tale oggetto è difettoso o meno.

A tal fine il thread *Main* deve creare due thread: *Machine* e *Quality*. *Machine* ogni 750 ms incrementa di uno il numero di prodotti creati (intero), produce un valore casuale tra -1000 e +1000 (float) e invia tali informazioni al thread *Quality*. In caso di float negativo il thread *Quality* deve stampare sullo standard output il messaggio "Errore " e poi il valore intero ricevuto, in caso di float positivo deve scrivere "OK". Per realizzare la comunicazione fra i thread *Machine* e *Quality* si utilizzi il costrutto PipedStream basato sugli oggetti serializzabili visto a lezione. Entrambi i thread devono estendere la classe Thread.

Inoltre, i thread *Main* e *Quality* comunicano tramite un oggetto condiviso e thread-safe chiamato *Overall*, che mantiene una sintesi degli oggetti con e senza difetti, La classe *Overall* deve esporre 4 metodi *incrementaDifetti()*, *incrementaCorretti()*, *getDifetti()* e *getCorretti()*. Il thread *Quality* utilizza i primi due metodi per tener traccia dei prodotti realizzati. Il thread *Main* dopo aver creato e avviato i *Machine* e *Quality* verifica ogni 1 s se il numero di oggetti difettosi è superiore a quelli corretti e, in tal caso, termina i due thread, ne attende la loro terminazione e infine termina.