

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

19 Giugno 2020 – durata 35’ per 10 CFU, 1h:10 per 12 CFU

10 e 12 CFU

Si realizzi un programma Java multithread che simuli il comportamento di un applicativo di sicurezza in esecuzione su un Server Web per il controllo di stringhe fornite in input dall’utente.

A tal fine il thread *Main* deve creare due thread, un primo thread chiamato *InputUtente* che riceve stringhe fornite in input dall’utente (via terminale) e un secondo thread *ScansionaInput* che analizza le stringhe lette da *InputUtente* per verificare la presenza di una eventuale stringa maligna. Entrambi i thread devono implementare l’interfaccia *Runnable*.

Per ogni stringa scansionata il thread *ScansionaInput* deve stampare a video un messaggio contenente la stringa “ok” nel caso in cui la stringa scansionata non contenga stringhe maligne e “pericolo” nel caso contrario.

Per realizzare la comunicazione fra i thread *InputUtente* e *ScansionaInput* si utilizzi il costrutto *PipedStream* bufferizzato visto a lezione. Inoltre, per semplicità, si considerino come stringhe maligne le stringhe “abcde” e “1234”.

Solo 10 CFU: Il thread *main* dopo aver creato e avviato i thread *InputUtente* e *ScansionaInput* si sospende per 30 secondi, poi termina i due thread, ne attende la loro terminazione e infine termina.

Da qui in poi solo 12 CFU

Oltre a quanto già specificato, il thread *ScansionaInput* e il thread *Main* comunicano tramite un oggetto condiviso e thread-safe chiamato *VerificaInput*.

Si deve quindi definire una classe *VerificaInput* che modella un “contatore”, che rappresenta il numero di stringhe sospette che sono state fornite in input dall’utente. La classe *VerificaInput* deve esporre due metodi *incrementaStringheSospette()* e *getNumeroStringheSospette()* per incrementare il valore del contatore e ricavarne il relativo valore.

Il thread *Main* si deve occupare di eseguire un controllo periodico (ogni 200 ms) su *VerificaInput* per verificare la sicurezza del sistema. Quando il thread *Main* legge un valore maggiore di 3 stringhe sospette si deve occupare di terminare i thread *InputUtente* e *ScansionaInput*, attendere la loro terminazione e infine terminare.

La soluzione deve essere thread-safe.