

PROGRAMMAZIONE CONCORRENTE

ESERCIZIO 1:

Un distributore di benzina con una sola pompa ha a disposizione 2000 litri di carburante.

Quattro automobili accedono una volta ogni secondo al distributore, si forniscono ogni volta di un numero casuale tra 20 e 50 litri di benzina.

Un'auto cisterna passa 15 secondi dopo l'inizio dell'esecuzione e prova a rifornire il distributore di benzina riportandolo a 2000 litri.

Il programma termina quando il carburante è terminato.

Usare il metodo `sleep` e `currentThread` per simulare la sosta al distributore e la stampa del nome del thread che in quel momento è in esecuzione.

Si implementi una soluzione in termini di threads e semafori.

Aggiungere le opportune stampe che verifichino il funzionamento del programma.

ESERCIZIO 2:

La cucina di un ristorante Fast-Food è gestito da un commesso e da un cuoco, che interagiscono attraverso uno scaffale, dove può essere deposto un solo ordine alla volta (l'ordine è composta da una bevanda e uno snack). Il cuoco prepara l'ordine in sequenza e lo depone sullo scaffale, attendendo che il commesso invii un ordine.

Scrivere un programma Java secondo il modello produttore-consumatore che gestisca la sincronizzazione e la mutua esclusione tra il cuoco e il commesso.

Il programma termina quando il cuoco comunica al commesso che il ristorante sta per chiudere.

Si implementi una soluzione in termini di threads e semafori.

Aggiungere le opportune stampe che verifichino il funzionamento del programma.