PCAD 24/25 - Laboratorio 5

1 Piscina

In questo esercizio, vi chiediamo di usare la classe `Semaphore` di JAVA che propone una implementazione dei semafori. Potete guardare online la documentazione di questa classe e in particolari le tre funzione seguente che rappresentano l'uso dei semafori visto a lezione:

- il costruttore `Semaphore(int permits)` per creare un nuovo semaforo inizializzato a `permits`;
- la funzione `void acquire()` che corrisponde al `wait` visto a lezione;
- la funzione `void release()` che corrisponde al `signal` visto a lezione.

Consideriamo una piscina pubblica con N_S spogliatoi individuali e N_C armadietti per lasciare i suoi vestiti. Un cliente che si presenta alla piscina realizza le tappe seguente:

(a) Prende la chiave di uno spogliatoio;

(b) Prende la chiave di un armadietto;

(c) Si cambia nello spogliatoio;

(d) Libera lo spogliatoio;

(e) Mette i suoi vestiti nel armadietto;

(f) Rida la chiave dello spogliatoio;

(g) Nuota (tenendosi la chiave del armadietto);

(h) Prende la chiave di un spogliatoio;

(i) Ricupera i suoi vestiti nel armadietto;

(j) Si riveste nello spogliatoio;

(k) Libera lo spogliatoio;

(l) Rida le chiave dello spogliatoio e del armadietto.

Lo scopo del esercizio è di proporre un implementazione in JAVA di un programma che simula il comportamento dei clienti in questa piscina.

1. Creare un programma JAVA che simula tanti clienti concorrenti. La disponibilità degli spogliatoi e degli armadietti sarà rappresentata da due semafori che saranno condivisi fra gli diversi thread che rappresentarono i clienti.
2. Testare il vostro programma con diversi numeri di clienti, di spogliatoi e di armadietti. In particolare, provate a trovare casi in cui c'è un deadlock (i.e. tutti i clienti sono bloccati).
3. Proponete un cambio minore nel comportamento dei clienti per evitare questo deadlock.

2 Cioccolatini

Consideriamo una scatola che può contenere al massimo P>0 cioccolatini. Abbiamo un pasticciere che in ciclo riempie la scatola con dei cioccolatini, ma la riempi direttamente con P cioccolatini e quindi lo può fare solo quando la scatola è vuota. Abbiamo anche dei mangiatori di cioccolatini che in ciclo prendono un cioccolatino dalla scatola e lo mangiano e se la scatola è vuota, aspettano che ci sia di nuovo cioccolatini.

Proponete in JAVA una modellazione di questo sistema, prendendo in considerazione che vogliamo un processo per il pasticciere e un processo per i mangiatori e che prenderemo tre mangiatori e P uguale a 5 e che vogliamo evitare le attese attive.

3 Consegna

Per la consegna, creare uno `zip` con tutti vostri file. Lo `zip` dovrà anche contenere un file `participanti.txt` dove gli nomi di chi ha participato alla consegna (questo anche se siete da solo a farla).