Nome: **<<Nome>>** Cognome: **<<Cognome>>** Compito: **<<Numero>>** Turno: **<<Turno>>**

PRINCIPI DI SISTEMI OPERATIVI
(A.A. 08-09)
28 Novembre 2008

IMPORTANTE:

1. Si considerano parte integrante delle soluzioni i **COMMENTI *significativi*** introdotti per facilitare la lettura del codice: come tali, essi influenzano la votazione finale. Tuttavia, i messaggi di debug (ad es. le println()) del programma NON SONO CONSIDERATI E QUINDI NON INFLUENZANO LA VOTAZIONE FINALE.
2. Il tempo a disposizione è di 90 minuti.
3. Il compito deve essere svolto solamente nel linguaggio Java, usando le classi del package **monitor** e lavorando con l'ambiente di sviluppo **IBM Eclipse**.
4. Seguire le seguenti regole per lo svolgimento dell'esame al laboratorio base:
   - *Fare il login in Linux con il proprio account.*
   - *Aprire Eclipse (comando "eclipse" da shell) e scegliere come workspace la propria home directory (es. /home1/n12345).*
   - *Creare, se non già presente, il progetto con le classi del monitor Java ed installare gli eventuali template presenti nella home.*
   - *Creare un progetto Java con nome "ESAME28Nov08-<<Turno>>-<<Numero>>" e scrivere le classi Java della soluzione nel package di default (senza nome) di tale progetto. Fare attenzione a scrivere correttamente il nome del progetto, con maiuscole e minuscole a posto!*
   - *Finito il vostro esame (o allo scadere del tempo di 1h:30m), dovete salvare tutto (si consiglia di salvare spesso per non perdere il proprio lavoro), chiudere Eclipse, fare il logout, lasciare il vostro PC e procedere alla consegna del testo.*

In una **pizzeria** da asporto lavorano *un solo* **pizzaiolo**, che prepara le pizze, ed *un solo* **fattorino**, che consegna le pizze a domicilio.

Il **pizzaiolo** lavora costantemente alla produzione delle pizze e può evadere un solo ordine di un cliente per volta. Le pizze possono essere ordinate da **clienti** *che vanno direttamente alla pizzeria (tipo C1)*, o **clienti** *che richiedono la consegna a domicilio e quindi ordinano per telefono (tipo C2)*. Per semplicità, ai fini della soluzione, si supponga che il tipo ed il numero di pizze siano ininfluenti: in altre parole, un cliente ordina genericamente "la pizza". Dopo aver preso un nuovo ordine, il pizzaiolo procede con la preparazione della pizza (*la cui durata si suppone variabile e random*). Una volta pronta la pizza:

1. se il cliente è di tipo C1 ritirata la pizza direttamente nella pizzeria, può pagarla e andare via.

2. se il cliente è di tipo C2, quando la pizza è pronta, deve attendere anche che vi sia un **fattorino** disponibile per la consegna a domicilio.

Il **fattorino**, se è libero, quindi parte su richiesta di un cliente di tipo C2 e viaggia verso l'abitazione del cliente (*il tempo del viaggio è variabile e random*). Una volta ricevuta la pizza a casa, il cliente la paga al fattorino e questi rientra infine in pizzeria: si supponga, sempre per semplicità, che il fattorino effettui sempre una consegna di un singolo ordine per volta!

Nella soluzione si garantisca la priorità ai clienti che ordinano per telefono (tipo C2) rispetto a quelli che vanno direttamente in pizzeria (tipo C1).

Si implementi una soluzione usando il costrutto monitor per modellare la **pizzeria** e i processi per modellare i **clienti**, il fattorino e il **pizzaiolo** e si descriva la sincronizzazione tra i processi. Nella soluzione si massimizzi l'utilizzo delle risorse (tenendo conto che pizzaiolo e fattorino lavorano indipendentemente). Si discuta se la soluzione proposta può presentare starvation e in caso positivo per quali processi, e si propongano modifiche e/o aggiunte per evitare la starvation.