Nome: **«Nome»** Cognome: **«Cognome»** Compito: **«Numero»** Turno: **«Turno»**

PRINCIPI DI SISTEMI OPERATIVI
(A.A. 12-13)
13 Febbraio 2013

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 (numero di tesserino e password di posta elettronica)
   - Aprire un terminale e digitare
     ```
     $ cd
     $ cd Desktop
     $ wget ftp://lica.lab.unimo.it/syncexam.sh
     $ chmod 755 ./syncexam.sh
     $ ./syncexam.sh
     ```
   - Aprire Elipse (comando "eclipse" sempre da shell)
   - Utilizzare come workspace la cartella "studente_xxxxx"
   - Creare un progetto Java con nome "ESAME130213-«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!
   - Installare le classi del monitor Java e gli eventuali template
   - Finito il vostro esame (o allo scadere del tempo), 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 un'**isola** vi è un **treno** che offre il giro dell'isola.
I **viaggiatori**, una volta che il treno è vuoto e in stazione, entrano per effettuare il giro. Questi possono viaggiare in 1° o 2° classe, a seconda del loro biglietto*.
Il treno ha N posti, 1/3 di prima classe e 2/3 di seconda classe.
Il treno parte all'orario prestabilito (scelto dal treno), effettua il giro dell'isola e una volta tornato alla stazione di partenza scarica i passeggeri che aveva precedentemente caricato, per poi ripartire per un nuovo giro all'orario prestabilito.
Prima della partenza il treno verifica il numero di posti liberi in prima classe, se ci sono dei posti disponibili e dei passeggeri col biglietto di seconda classe che non sono potuti salire sul treno (per mancanza dei rispettivi posti in seconda classe), fa salire questi ultimi per riempire il più possibile il treno.

Si implementi una soluzione usando il costrutto monitor per modellare l'**isola**, i processi per modellare i **viaggiatori** e il **treno**. Nella soluzione si massimizzi l'utilizzo delle risorse. 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.

* Nel main si modellino i clienti perché almeno 1/5 di essi abbia il biglietto di prima classe