

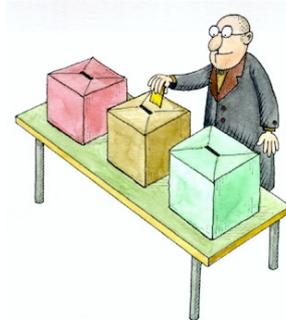
# Sistemi Operativi T

## Prova di laboratorio

### 30 Giugno 2015

#### TEMA A

#### 1. Esercizio di Programmazione Concorrente in Java



Sono indette le votazioni per l'elezione del Rettore dell'Ateneo di Bologna. A questo scopo è stato allestito un seggio all'interno di un'aula, dove ogni elettore può esprimere il proprio voto.

Gli elettori sono suddivisi in tre categorie:

1. docenti (PR)
2. personale tecnico-amministrativo (TA)
3. rappresentanti degli studenti (ST)

Il seggio contiene:

- N cabine elettorali. Ogni cabina può ospitare un elettore (di qualunque tipo) alla volta. Per evitare affollamento, il seggio ha pertanto una capacità limitata pari a N, che esprime il massimo numero di elettori che possono stare contemporaneamente al suo interno.
- 3 registri (ognuno gestito da un membro della commissione elettorale) per le firme di presenza degli elettori; in particolare:
  - 1 registro per i PR
  - 1 registro per i TA
  - 1 registro per gli ST

Per evitare attese all'interno del locale, un elettore non può entrare se c'è un altro elettore della stessa categoria che sta firmando il registro.

Pertanto, ogni elettore ha il seguente comportamento:

- **entra** nel seggio (dopo un'eventuale attesa) e si reca dal commissario dedicato alla propria categoria per la **firma** del registro;
- **libera il commissario e occupa una cabina** elettorale per il voto;
- **esce** dalla cabina, deposita la scheda nell'urna ed esce dal seggio.

Realizzare un'applicazione **concorrente** in Java basata sul monitor nella quale **gli elettori** siano rappresentati da **thread distinti**.

Si tenga presente che sia l'operazione di firma che l'operazione di voto possono richiedere tempi non trascurabili.

La politica di ammissione degli elettori al seggio dovrà tenere conto di tutti i vincoli indicati ed inoltre dare la priorità agli elettori seguendo l'ordine:

1. ST
2. TA
3. PR