

Esercizio 4: Traccia CasaDiCura (I)

- Si consideri una casa di cura specializzata in operazioni in “day ospital”. La casa di cura è composta da 2 sale: una piccola sala di preparazione in grado di ospitare fino a 3 pazienti, e una sala delle operazioni gestita da un medico.
- Tra i pazienti in attesa dell’operazione, solo in 3 possono entrare nella sala di preparazione (risveglio casuale). I pazienti presenti nella sala di preparazione verranno chiamati uno per volta dal medico (risveglio FIFO) per entrare nella sala delle operazioni. L’operazione del medico dura un tempo random tra 20 e 40 minuti. Ad operazione conclusa, il medico permette al paziente di uscire (il paziente abbandona la sala delle operazioni, e quindi un altro paziente può entrare nella sala di preparazione). Il medico prepara la sala per una nuova operazione impiegando 20 minuti.

Esercizio 4: Traccia CasaDiCura (II)

- Si modelli il sistema descritto in Java, dove Paziente e Medico sono dei Thread che interagiscono tramite un oggetto CasaDiCura che espone solo i seguenti metodi:
 - void **chiamaEIniziaOperazione()**: il medico chiama un nuovo paziente e attende che il paziente sia entrato in sala operatoria. Dopo esserci assicurato che il paziente è in sala operatoria inizia l'operazione.
 - void **fineOperazione()**: il medico segnala al paziente che l'operazione è conclusa e che può lasciare l'ospedale. Non deve attendere l'uscita del paziente dalla sala operatoria.
 - void **pazienteEntra()**: sospende il paziente fin quando non entra in sala operatoria.
 - void **pazienteEsci()**: segnala al paziente che ha concluso l'operazione e che può uscire dalla sala operatoria.
- Si implementi la classe **CasaDiCura** (astratta), **Paziente** e **Medico**, e una soluzione che riproduca il funzionamento del problema sopra descritto utilizzando la classe Semaphore (usare solo i metodi acquire e release) del package java.util.concurrent (**CasaDiCuraSem**). Si crei un main che esegue il **Dottore** e ogni minuto crea e lancia un nuovo Paziente.
- Si implementi anche una soluzione che riproduca il funzionamento del problema sopra descritto utilizzando gli strumenti di mutua esclusione e sincronizzazione del package java.util.concurrent.locks (**CasaDiCuraLC**).

Esercizio 4: Traccia CasaDiCura (III)

- Considerare la seguente variante. Nella casa di cura sono presenti diverse sale operatorie e diversi medici. Commentare, anche attraverso l'uso di frammenti di codice, come bisogna modificare le classi definite precedentemente per permettere a diversi medici di eseguire operazioni in parallelo.