TECNICHE DI PROGRAMMAZIONE

28/05/19

Continuazione esercizio “Emergency”.

Il professore ha messo a posto alcuni aspetti riguardanti l’interfaccia del programma in modo da renderla più facile da comprendere. Nel nostro caso, la simulazione non si appoggia su dati quindi il simulatore può essere creato direttamente nel Main. Ci sono però casi più difficili in cui non possiamo fare così. In questo caso, avremo una classe Model e il Simulatore verrà creato nel Model solo quando serve.

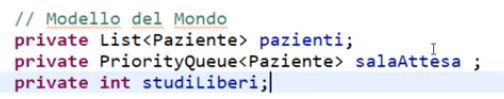
Andiamo a gestire le sale studio mediche. Abbiamo 2 strade possibili:

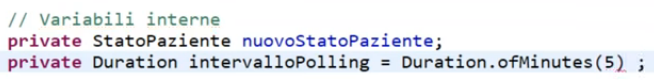
1. Creare un meccanismo automatico per aggiornare lo stato del mondo. Possiamo creare quindi un evento che si aggiorna periodicamente e che ha come obiettivo osservare lo stato del mondo. Sarà un evento che aggiungerò alla coda degli eventi il cui unico scopo è di andare a leggere lo stato, vedere se è verificata una certa condizione e. in caso affermativo, schedulare un evento.
2. Chiedersi come fa il sistema ad arrivare nella condizione a noi favorevole (lista di attesa non vuota e almeno uno studio medico libero). Cerchiamo di trovare tutti i modi possibili per arrivare a questa condizione e andiamo ad intercettare tutti gli eventi che l’hanno causata. È più difficile rispetto al precedente in quanto dobbiamo considerare tutti i casi possibili.

Decidiamo di iniziare dal metodo 1. Bisogna inserire un osservatore (l’infermiere) che ogni tot di minuti entra e controlla lo stato del mondo. Questo osservatore non è altro che un tipo di evento nuovo (POLLING) e periodico usato per verificare se ci sono studi liberi e pazienti in attesa. Aggiungiamo anche una durata tra ogni osservazione di 5 minuti.

Dopo di che aggiungiamo il primo evento in cui viene lanciato l’osservatore in polling.

Come ultima cosa andiamo a gestire il case POLLING in cui verifichiamo se ci sono pazienti in attesa con studi liberi e in cui rischeduliamo l’evento creato prima.







Per verificare se ci sono pazienti in attesa, decidiamo di creare una “lista di attesa” di tipo “priorityqueue”. La inizializziamo nel metodo init().

Per quanto riguarda la priorità abbiamo già detto che dobbiamo dare un metodo attraverso il quale comparare diversi oggetti. Questo oggetto lo chiamiamo “prioritàPaziente”. Andiamo a creare questa classe che dovrà implementare l’interfaccia Comparable.

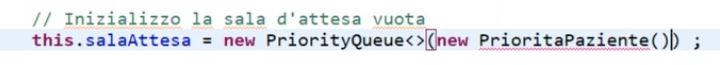


Immagine che contiene screenshot

Descrizione generata automaticamente

Quindi, il case POLLING sarà il seguente:

Immagine che contiene screenshot

Descrizione generata automaticamente

Andiamo quindi a gestire il case VISITA. Creiamo un oggetto paziente “pazChiamato” che conterrà il primo paziente disponibile dalla lista di attesa. Quando il paziente viene chiamato ed entra nella sala studio, il suo stato passerà a TREATING (lo stanno trattando). Lo studio sarà occupato quindi diminuisco di uno gli studi liberi. Come ultima cosa, schedulo l’uscita del paziente.

Immagine che contiene screenshot

Descrizione generata automaticamente

Andiamo infine a gestire lo stato CURATO. Dobbiamo indicare che il paziente è fuori (è stato dismesso dopo la cura. Bisogna aggiornare il numero di persone dismesse e il numero di studi liberi. Come ultima cosa chiamiamo l’evento VISITA in quanto sappiamo che è presente uno studio libero e quindi non ci seve il POLLING (l’infermiera che passa ogni volta per controllare) per verificare che sia libero.

Immagine che contiene screenshot

Descrizione generata automaticamente

Andiamo a modifica l’evento VISITA nel caso in cui non ci siano più pazienti che attendono di essere curati. In tal caso dobbiamo uscire da VISITA attraverso un break.

Immagine che contiene screenshot

Descrizione generata automaticamente

Il POLLING ci serve solo alle 8 di mattino quando le sale studio solo tutte libere per vedere se ci sono pazienti da visitare e nel caso in cui un paziente di una sala studio viene curato troppo velocemente e non ci sono pazienti in attesa. In tutti gli altri casi, quando un paziente viene curato, si controllano se ci sono pazienti fuori che hanno bisogno di cura e, in caso positivo, viene visitato quello con priorità più alta.

Runnando il programma notiamo che il POLLING va avanti all’infinito. Impostiamo quindi un tempo di fine in modo tale da evitare questo problema.

Immagine che contiene screenshot

Descrizione generata automaticamente

Ora come ora il programma funziona ma bisogna ancora attuare delle migliorie. Modifichiamo il TIMEOUT. Come prima cosa, se un paziente arriva al Timeout, decidiamo di toglierlo dalla lista SalaAtteasa.

Lo andiamo ad aggiungere successivamente nel “RED” dopo che gli ho cambiato lo stato. Così facendo, il paziente si troverà ad una posizione migliore e verranno visitati prima i pazienti rossi degli altri.

Il programma è terminato ed è leggermente più difficile rispetto a quelli che avremo all’esame.

FINE