

## Ospedale

In un Ospedale dove lavorano **N** medici arrivano dei **pazienti**. A ciascun paziente, non appena arriva all'ospedale, viene assegnato un codice a seconda della sua priorità: **rosso** per i casi molto gravi, **giallo** per i casi meno gravi, **verde** per i casi non gravi.

I pazienti devono essere curati da un medico (durata random) seguendo le priorità:

1. i pazienti con codice verde hanno la priorità più bassa e devono essere serviti solo se non ci sono altri pazienti con codici più gravi in attesa.
2. i pazienti con codice giallo hanno priorità maggiore di quelli con codice verde e devono essere serviti solo se non ci sono pazienti in codice rosso.
3. i pazienti con codice rosso hanno la priorità maggiore di tutti e devono essere serviti subito.

Finite le visite, i pazienti vengono dimessi dall'ospedale e possono andare via.

Si modelli e si risolva il problema usando i costrutti nativi di Java, modellando Pazienti e Dottori come Thread e tenendo in considerazione che:

1. qualunque medico è in grado di curare chiunque, cioè non esistono pazienti che necessitano di un medico specifico.
2. per semplicità, un medico finisce sempre il proprio paziente prima di passare al successivo, cioè se un medico sta servendo un paziente, anche se arriva un paziente con maggiore priorità, quest'ultimo deve aspettare che il medico abbia finito con il suo paziente, e poi può essere servito.
3. i thread pazienti devono rimanere alive finché il dottore non ha terminato di curarli: solo alla fine delle cure possono terminare.

## Aggiunta

Gestire i pazienti in modo FIFO-like, cioè i pazienti con lo stesso livello di priorità vengono serviti seguendo il loro ordine di arrivo in ospedale.