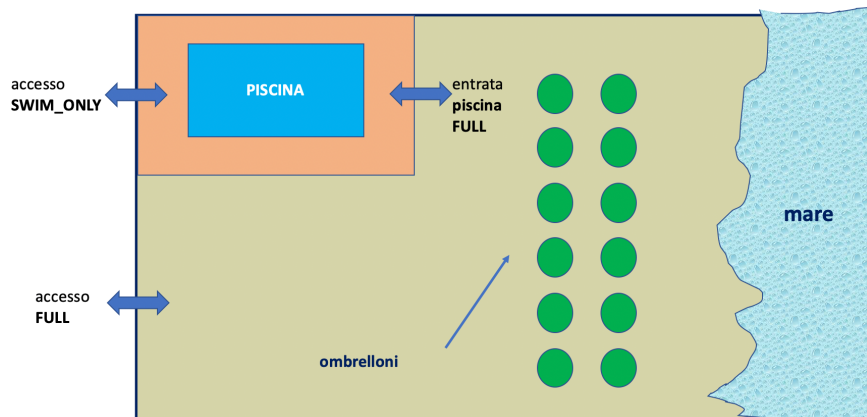


Sistemi Operativi T

Prova di laboratorio

13 Luglio 2021

Esercizio di Programmazione Concorrente in Java



Si consideri uno stabilimento balneare in una località marittima, suddiviso in **due aree distinte**:

- la **spiaggia** attrezzata con ombrelloni e sdraio;
- una **piscina**, situata in posizione arretrata rispetto alla spiaggia

I **clienti dello stabilimento balneare** possono essere di due tipi:

- clienti "SWIM_ONLY" che usano solo la PISCINA, accedendo da un'entrata dedicata SWIM_ONLY, esterna allo stabilimento (v. figura);
- clienti "FULL" che usufruiscono della SPIAGGIA e della PISCINA: ogni cliente di questa categoria **occupa un ombrellone** e, una volta entrato nello stabilimento tramite l'accesso FULL (v. figura), può spostarsi tra le 2 aree. Nel caso voglia andare in piscina, utilizzerà l'entrata piscina FULL (v. figura); quando il cliente FULL è in piscina, il suo ombrellone rimane occupato. I clienti FULL possono essere di 2 tipi:
 - abbonati: hanno acquistato un abbonamento stagionale
 - occasionali: accedono allo stabilimento occasionalmente

La spiaggia è attrezzata con N_OMB ombrelloni, pertanto un cliente FULL potrà entrare nello stabilimento solo se c'è un ombrellone libero.

L'area della PISCINA ha una capacità limitata fissata rispettivamente a Np (massimo numero di clienti ammessi in piscina): l'accesso alla piscina di ogni cliente (cliente SWIM_ONLY o cliente FULL) dovrà essere sempre subordinato al rispetto del vincolo di capacità.

Pertanto, **il comportamento tipico di un cliente SWIM_ONLY** è il seguente:

1. **Entra in piscina** tramite l'entrata esterna;

<rimane in piscina per un tempo arbitrario>
2. **Esce dalla piscina** tramite l'entrata SWIM_ONLY

Sistemi Operativi T

Prova di laboratorio

13 Luglio 2021

Il comportamento tipico di un cliente FULL è, invece, il seguente:

1. **Entra nello stabilimento occupando un ombrellone** tramite l'entrata FULL;

Ripete per un numero arbitrario di volte: {

{ <sta in spiaggia per un tempo arbitrario>

Se **decide** di andare in piscina:

{ 1.a **Entra in piscina** dall'entrata piscina FULL (il suo ombrellone rimane occupato)

<rimane in piscina per un tempo arbitrario>

1.b **Esce dalla piscina** tramite l'entrata piscina FULL

}

}

2. **Esce dallo stabilimento** tramite l'accesso FULL

Realizzare un'applicazione in **Java** basata su monitor e variabili condizione, nella quale i **clienti**, siano rappresentati da **thread concorrenti**. La sincronizzazione tra i processi dovrà tenere conto dei vincoli dati ed, inoltre, dei seguenti vincoli:

- **nell'ingresso dei clienti FULL allo stabilimento**, i clienti abbonati siano privilegiati rispetto agli occasionali;
- **nell'ingresso alla piscina**: i clienti SWIM_ONLY abbiano la precedenza sui clienti FULL; tra i clienti FULL gli abbonati abbiano la priorità sugli occasionali.