

## Verifica monitor: coda prioritaria

La verifica consiste nell'implementazione di un monitor **CodeMonitor** per la sincronizzazione di un certo numero di thread **Persona**. I thread **Persona** rappresentano gli utenti di un servizio con uno sportello e **due code**: una **normale** e una **prioritaria**. Le persone, in base al proprio status (**verde**=prioritarie, **blu**=normali), si mettono in fila sulla rispettiva coda.

Gli utenti della coda prioritaria hanno la **precedenza** sugli altri, ma solo se sono arrivati alla fine della coda e sono effettivamente in attesa del servizio (se sono in movimento e si stanno accodando non hanno precedenza!).

Un thread **Persona** si comporta come segue:

```
Thread Persona(boolean priority) {
    // si accoda e procede fino alla fine della coda

    // è alla fine della coda attende lo sportello
    sportello = p.attendiSportello(priority);

    // va allo sportello e attende il servizio

    // libera lo sportello e esce
    p.liberaSportello(sportello);
}
```

I metodi del monitor che dovete implementare sono i seguenti:

```
// La persona è la prossima ad essere servita e attende che si liberi
// lo sportello. Se la persona è sulla coda prioritaria (priority è
// true) appena lo sportello si libera la persona può procedere. Se
// la persona è sulla coda normale (priority è false) la persona
// attende che non ci siano altre persone in coda prioritaria già
// in attesa e che lo sportello si liberi. In altre parole, dà
// la precedenza a persone in attesa in coda prioritaria.
```

```
synchronized void attendiSportello(boolean priority)
    throws InterruptedException {
    if (priority) {
        // gestione coda prioritaria
    } else {
        // gestione coda non prioritaria
    }
}

// La persona ha raggiunto lo sportello, ha fruito del servizio e ora
// lo libera
```

```
public synchronized void liberaSportello() {
}
```

Scaricate l'allegato e modificate il file **CodeMonitor.java**. Quando un thread **Persona** viola le regole delle code diventa rosso e viene stampato un messaggio di errore (**Attenzione**: il test non è esaustivo, se non dà messaggi di errore non è detto che il codice sia corretto!).

Per compilare ed eseguire il programma, usate i seguenti comandi:


```
> javac Code.java
> java Code
```

### NOTA BENE

- Consegnate solo il file **CodeMonitor.java**;
- **Commentate** in maniera appropriata il vostro programma: in particolare, inserite un commento iniziale in cui spiegate l'idea risolutiva. Soluzioni non commentate **NON** saranno valutate.
- **NON COPIATE!!!**

 [Code.zip](#)

## Stato consegna

Stato consegna	Consegnato per la valutazione
Stato valutazione	Valutata
Termine consegne	giovedì, 23 aprile 2020, 15:30
Tempo rimasto	Il compito è stato consegnato 39 min. 30 secondi in anticipo
Ultima modifica	giovedì, 23 aprile 2020, 14:50
Consegna file	 <a href="#">CodeMonitor.java</a>
Commenti alle consegne	 <a href="#">Commenti (0)</a>

## Commento

Valutazione	2,00 / 3,00
Data di valutazione	venerdì, 1 maggio 2020, 18:46
Valutatore	 DILETTA OLLIARO
Commenti	Mancano i commenti nel codice.

◀ [Verifica semafori: Imbuto 2.0](#)

Vai a...

