

Tempo a disposizione: 2 ore

Risultati: 11 febbraio 2022, dopo le 19, via e-mail

Discussione degli elaborati ed eventuale prova orale: la data verrà comunicata via e-mail

Si consideri un sistema software S che contenga degli oggetti corridor che implementano la seguente interfaccia:

```
package it.unipr.informatica.exam;

public interface Corridor {
    public void enter();
    public void exit();
}
```

Un oggetto corridor ha il seguente comportamento:

1. Tutte le volte che viene chiamato `enter()`, il chiamante si blocca.
2. Se il numero di chiamate bloccate raggiunge o supera $K = 5$, vengono sbloccate K chiamate.
3. Le chiamate sbloccate proseguono e prima o poi verrà chiamato `exit()` da ogni chiamata sbloccata.
4. Se ci sono ancora almeno K chiamate bloccate, si ritorna al punto 2 per sbloccare altre K chiamate.

Quindi, un oggetto corridor si comporta come un corridoio che viene attraversato esattamente da K chiamate in modo concorrente. Una chiamata entra nel corridoio con `enter()` ed esce dal corridoio con `exit()`. Si noti che il comportamento degli oggetti corridor non sono definiti se viene chiamato `enter()` ma poi non viene chiamato un relativo `exit()`. Similmente, il comportamento non è definito se viene chiamato `exit()` senza una precedente chiamata ad `enter()`.

Nel sistema S sono presenti $W = 50$ worker che vengono creati da un singleton detto worker manager a cui è associato un oggetto corridor. Ogni worker implementa la seguente interfaccia:

```
package it.unipr.informatica.exam;

public interface Worker {
    public void execute();
    public Corridor getCorridor();
}
```

Il metodo `getCorridor()` consente di accedere all'oggetto corridor associato al worker manager. Il metodo `execute()` ha il seguente comportamento:

1. Chiama `enter()` sull'oggetto corridor associato al worker manager.
2. Stampa il messaggio In corridor seguito dall'identificativo del thread corrente ottenuto mediante `Thread.currentThread().getId()`.
3. Chiama `exit()` sull'oggetto corridor associato al worker manager.

Realizzare il sistema software S in Java aggiungendo, per avviare il sistema, la classe `Main` nel package `it.unipr.informatica.exam`. Il metodo principale della classe `Main` crea e attiva un thread per ogni worker gestito dal worker manager. Il thread relativo ad un worker si limita a chiamare `execute()` sul proprio worker e a stampare il messaggio Done seguito dall'identificativo del thread corrente.

(segue nel retro del foglio)