

Programmazione Concorrente e Parallela - Serie 6

Maura Clerici

05.04.2019

Esercizio 2

Eseguendo il programma, impiega circa 18 secondi per terminare. È ottimizzabile analizzando il metodo `occupaBagno()`. È stata quindi tolta la definizione `synchronized`:

```
public boolean occupaBagno(final boolean uomo) {...}
```

Successivamente, sono stati definiti `synchronized` i metodi `provaOccupare()` e `libera()` della classe `Bagno`:

```
class Bagno {
    private boolean occupato = false;

    // Ritorna false se già occupato
    public synchronized boolean provaOccupare() {
        if (occupato)
            return false;
        this.occupato = true;
        return true;
    }

    public synchronized void libera() {
        this.occupato = false;
    }
}
```

Nella soluzione iniziale, il programma impiega molto tempo perché il metodo `occupaBagno()` blocca l'accesso anche ai bagni liberi della collection. Nella versione ottimizzata tale responsabilità viene presa dalla singola istanza `Bagno`, smettendo di bloccare le altre istanze. Questa soluzione impiega circa 4 secondi per terminare.