

BlockingQueue

Friday, 17 March 2023 09:50

Ho fatto un piccolo approfondimento su questo argomento: [Java BlockingQueue](#)

- Aggiungiamo elementi in una coda, thread, e poi li togliamo
- La differenza è che, se la coda è piena, visto che la lunghezza della coda è fissata, l'aggiunta verrà messa in attesa
- Quando un thread uno dei nostri thread di sistema è libero, cerca di prendere un thread nella queue se disponibile
- E' una interfaccia, abbiamo:
 - ArrayBlockingQueue
 - LinkedBlockingQueue
- Questo è thread friendly

```
@Override
public void run() {
    while(true) {
        long timeMillis = System.currentTimeMillis();
        try {
            this.blockingQueue.put("" + timeMillis);
        } catch (InterruptedException e) {
            System.out.println("Producer was interrupted");
        }
        sleep(1000);
    }
}
```

```
while(true){
    try {
        String element =
            this.blockingQueue.take();
        System.out.println("consumed: " + element);
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
}
```

- Inserire:
 - Put dove aggiungiamo, se però la coda è piena veniamo messi ad aspettare
 - Add, se non c'è spazio non verremo messi ad aspettare, però lancia un IllegalStateException
 - Offer, da true/false se è stato aggiunto con successo
 - Noi possiamo metterci un timeout dove, se la queue è piena proveremo ad aspettare X millisecondi per cercare di rimetterlo
- Prendere:
 - Take, è praticamente un put dove noi aspettiamo
 - Poll, ritorna il primo elemento ma se nulla c'è ritorna null
 - Possiamo aggiungerci un timeout come in offer
 - Remove, dove noi rimuoviamo se esiste, nota: rimuoviamo l'elemento, non GLI elementi e se non esiste ritorna false
 - DrainTo
 - Prendiamo tutti gli elementi dalla queue
 - Possiamo aggiungerci un numero per determinare il numero massimo
 - Peek, dove noi prendiamo il primo elemento senza rimuoverlo, ritorna null
 - Element, ritorna il primo elemento però se non c'è elemento ritorna NoSuchElementException
- Altri metodi:
 - Size, quanti elementi abbiamo
 - RemainingCapacity, quanti elementi è possibile mettere
 - Contains, questo è auto esplicativo

