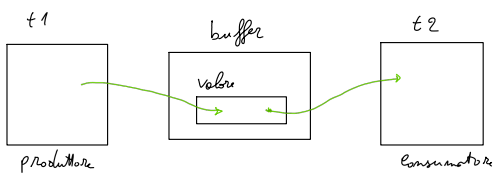
Lezione 11

martedì 13 novembre 2018 14:33

## wait, notify, notifyAll

- synchronized: permette di ottenere mutua esclusione e risolvere i problemi di interferenza tra flussi di esecuzione
- wait, notify, notifyAll: permettono di imporre vincoli di ordinamento temporale (il programmatore vuole che una certa operazione venga prima/dopo un'altra)

Esempio:

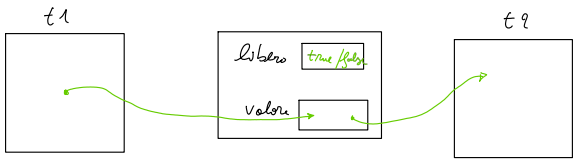



Vogliamo che nessun valore vada perso

Come può t1 capire se t2 ha consumato il valore precedente?

Come può t2 capire se nel buffer c'è un nuovo valore?

Soluzione (sbagliata):




t1:
- attende libero == true
- scrive nuovo valore
- libero = false

t2:
- attende libero == false
- estrae il valore
- libero = true

Problema: t1 e t2 eseguono attese attive (busy wait)

Soluzione corretta: usare wait e notify

- wait:
  - il thread che invoca wait() sospende la propria esecuzione
  - il thread viene inserito nel wait-set dell'oggetto su cui è stata eseguita la wait
  - per poter chiamare la wait su un oggetto il thread deve possedere il lock su tale oggetto
  - il lock viene rilasciato nel momento in cui il thread si blocca sulla wait; viene ri-acquisito automaticamente quando riprende l'esecuzione del thread
- notify:
  - invia una notifica a un thread a caso tra quelli del wait-set su cui la notify viene invocata
  - il thread selezionato esce dallo stato bloccato e torna in coda pronti
  - il thread ri-acquisisce il lock (eventualmente competendo con altri thread interessati allo stesso lock).
  - per chiamare notify su un oggetto è necessario

avere il lock su tale oggetto

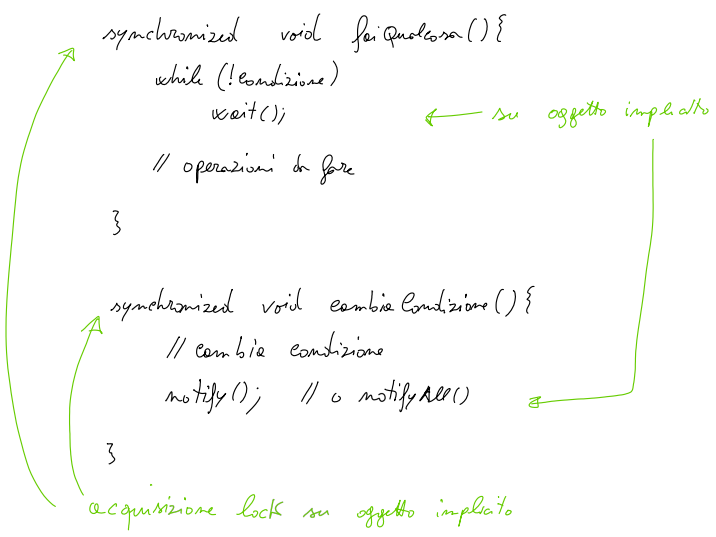
- notifyAll:
  - come la notify, ma agisce su tutti i thread nel wait-set
  - in particolare: tutti i thread nel wait-set tornano in coda pronti; tutti tentano di acquisire nuovamente il lock; uno lo conquista, gli altri si bloccano in attesa che il lock torni libero

Se un thread invoca wait(), notify() e notifyAll() senza avere il lock sull'oggetto su cui il metodo è invocato:

viene generata IllegalMonitorStateException

Normalmente notify, notifyAll, wait vengono invocati sull'oggetto implicito
- la classe definisce le politiche di sincronizzazione e mutua esclusione

Forma normalmente usata:




- Se viene fatta una notify o una notifyAll su un oggetto e se il wait-set di tale oggetto è vuoto (non ci sono thread bloccati su tale oggetto) la notify (o la notifyAll) non ha effetto (si perde).
  (Non c'è "memoria")

- notify o notifyAll?
  - più thread possono essere bloccati sullo stesso oggetto.
  - se i thread possono bloccarsi su condizioni diverse è necessario usare notifyAll
    - se esegue notify posso risvegliare un thread bloccato su una condizione diversa da quella che è appena diventata vera
    - il thread risvegliato trova la condizione ancora falsa e si ribblocca (deadlock)

Usare notify è un'ottimizzazione possibile se
- tutti i thread sono in attesa della stessa condizione
- uno solo dei thread può trarre beneficio dal fatto che la condizione sia vera.