

SISTEMI OPERATIVI IIN/IEL/IDT
INFORMATICA INDUSTRIALE E SISTEMI OPERATIVI IDI
SISTEMI DI ELABORAZIONE P.O.
prova scritta del 29.04.2005

Nome: _____

Cognome: _____

Un servizio di sportello al pubblico (es. segreteria studenti) opera nel modo seguente:

- Gli utenti arrivano in modo casuale, entrano in una sala d'attesa dove acquisiscono un numero univoco progressivo e si sospendono in attesa di poter essere serviti;
- Tre lavoratori svolgono il servizio di sportello gestendo il turno degli utenti attraverso un display numerico condiviso che indica il prossimo numero da servire. In assenza di utenti da servire i lavoratori rimangono sospesi;
- Gli utenti ad ogni aggiornamento del display verificano il numero in loro possesso con quello visualizzato e, se possibile, si presentano allo sportello per essere serviti;
- Al termine del servizio gli utenti lasciano il sistema.

Ponendo utenti e lavoratori in corrispondenza con due diversi tipi di thread si descriva la soluzione Java che permetta la sincronizzazione tra utenti e lavoratori.

Nella soluzione si privilegi la schematizzazione dei meccanismi di sincronizzazione rispetto ad altri dettagli di implementazione.

Soluzione

```
public class Main {
    public static void main(String[] args) {
        // oggetti condivisi tra utenti e lavoratori
        Distributore d = new Distributore();
        Pannello p = new Pannello();

        for (int i=0;i<3;i++)
            (new Lavoratore("L-" + i, d, p)).start();
        for (int j=0;j<50;j++) {
            (new Utente("U-" + j, d, p)).start();
            try {
                Thread.sleep((int)(3.0*Math.random()*1000));
            }
            catch(InterruptedException ie) {
            }
        }
    }
}

// classe che modella il numero acquisito dagli utenti
public class Distributore
{
    // ultimo numero erogato
    private int numero = 0;

    // assegna il prossimo numero all'utente che ne fa richiesta
    public synchronized int nuovoNumero() {
        numero ++;
        notify();
        return numero;
    }

    public synchronized int ultimoNumero() {
        return numero;
    }
}

// classe che modella il display aggiornato dai lavoratori
public class Pannello
{
    // ultimo numero servito
    private int chiamato = 0;

    public synchronized int prossimo() {
        chiamato ++;
        notifyAll();
        return chiamato;
    }

    public synchronized boolean confronta(int numero) {
        // se l'utente è già stato chiamato può considerarsi servito
        return (numero<=chiamato);
    }
}
```

```

// classe che modella i lavoratori agli sportelli che servono i clienti in ordine di numero
public class Lavoratore extends Thread
{
    // display utilizzato per indicare il prossimo utente da servire
    private Pannello p;

    // distributore di numeri su cui il lavoratore si sospende se inattivo
    private Distributore d;

    public Lavoratore(String n, Distributore d, Pannello p) {
        super(n);
        this.p = p;
        this.d = d;
    }

    public void run() {
        while (true) {
            synchronized(d) {
                try {
                    if (p.confronta(d.ultimoNumero())) {
                        System.out.println("Lavoratore " + getName() + "
inattivo");
                        d.wait();
                    }
                    int n = p.prossimo();
                    System.out.println("Lavoratore " + getName() + " eroga
servizio all'utente " + n);

                    // serve utente
                    // .....
                }
                catch (InterruptedException ie) {
                }
            }
        }
    }
}

```

```

// classe che modella gli utenti in arrivo che acquisiscono il numero prima di essere serviti
public class Utente extends Thread
{
    // pannello utilizzato per indicare il prossimo utente da servire
    private Pannello p;

    // distributore che eroga il numero progressivo
    private Distributore d;

    public Utente(String n, Distributore d, Pannello p) {
        super(n);
        this.p = p;
        this.d = d;
    }
}

```

```
public void run() {
    System.out.println( "Utente " + getName() + " chiede numero " );
    int n = d.nuovoNumero();
    boolean servito = false;
    System.out.println("Utente " + getName() + " chiede servizio, con numero " + n);

    while (!servito) {
        synchronized(p) {
            try {
                p.wait();
                servito = p.confronta(n);
            }
            catch(InterruptedException ie) {
            }
        }
    }
    System.out.println( "Utente " + getName() + " ha ottenuto il servizio" );
}
}
```