

SISTEMI OPERATIVI IIN/IEL/IDT

INFORMATICA INDUSTRIALE E SISTEMI OPERATIVI IDI

SISTEMI DI ELABORAZIONE P.O.

prova scritta del 06.09.2005

Nome: _____

Cognome: _____

In una fabbrica lavorano M operai che assemblano motociclette. Ogni volta che un operaio ha completato una motocicletta va ad aggiornare un contatore che tiene traccia del numero di motociclette prodotte complessivamente dalla fabbrica.

Supponendo che un operaio possa essere rappresentato da un thread, si definisca una classe Java che riproduca il comportamento di un operaio. Si definisca inoltre una classe Java che periodicamente legge il contatore di cui sopra e ne visualizza il valore. Infine, si realizzi un programma che istanzia ed avvia i threads corrispondenti agli M operai ed il thread che visualizza il numero di motociclette assemblate.

Soluzione

```
public class Contatore {
    public void incrementa() { valore++; }
    public int valore() { return valore; }
    private int valore = 0;
}

public class Operaio extends Thread {
    public Operaio( Contatore c ) { contatore = c; }
    public void run() {
        while ( true ) {
            /*
             * ... assembla motocicletta ...
             */
            synchronized( contatore ) {
                contatore.incrementa();
            }
        }
    }
    private Contatore contatore;
}

public class Visualizzatore extends Thread {
    public Visualizzatore( Contatore c ) { contatore = c; }
    public void run() {
        while ( true ) {
            try {
                Thread.sleep( 1000 );
            }
            catch ( InterruptedException ie ) {}
            int v;
            synchronized( contatore ) {
                v = contatore.valore();
            }
            System.out.println( "valore: " + v );
        }
    }
    private Contatore contatore;
}

public class Main {
    public static final int M = 10;
    public static void main( String[] args ) {
        Contatore c = new Contatore();
        for ( int i = 0; i < M; i++ ) {
            Operaio o = new Operaio( c );
            o.start();
        }
        Visualizzatore v = new Visualizzatore( c );
        v.start();
    }
}
```