

Programmazione in rete e laboratorio

Secondo esonero – 10 dicembre 2004

1. La classe **JSlider** realizza un cursore (*slider*).



Il costruttore **JSlider(min, max, init)** costruisce un cursore i cui valori sono compresi fra *min* e *max* e che ha un valore iniziale *init*. Il metodo **getValue()** restituisce un intero che fornisce il valore del cursore. Quando l'utente sposta il cursore, viene generato un **ChangeEvent**. Le librerie di Java forniscono l'interfaccia **ChangeListener** che ha un unico metodo **stateChanged**.

Data la seguente classe:

```
class ProvaSlider extends JFrame
{   JSlider slider;
    JLabel label;
    JPanel panel;

    public ProvaSlider()
    {   slider = new JSlider(0,30,15);
        label = new JLabel("    ");
        panel = new JPanel();
        panel.add(slider);
        panel.add(label);
        getContentPane().add(panel);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        pack();
        setVisible(true);
    }
}
```

completarla in modo che la *label* contenga sempre il valore corrente del cursore.

2. Supponendo che un *thread* sia in possesso del *lock* di un oggetto, spiegare in quali dei casi elencati il *thread* rilascia il *lock*:

- a) quando esegue *notify*
- b) quando esegue *notifyAll*
- c) quando si mette in *wait*
- d) quando esegue la *sleep*
- e) quando esce da un blocco sincronizzato

3. Date le seguenti classi

```
class Contatore
{   private int val = 0;
    public void incr()
    {   int temp = val;
        temp++;
        val = temp;
    }
    public int getVal()
    {   return val;}
}

class IncrThread extends Thread
{   Contatore cont;
    public IncrThread(Contatore c)
    {   cont = c;}
    public void run()
    {   cont.incr();}
}

class Prova
{   public static void main(String[] args)
    {   Contatore c = new Contatore();
        IncrThread t1 = new IncrThread(c);
        IncrThread t2 = new IncrThread(c);
        t1.start();
        t2.start();
    }
}
```

- a) Spiegare perché il main di Prova potrebbe non funzionare correttamente.
- b) Modificare il programma in modo che il contatore venga incrementato correttamente.

4. Si consideri il problema dei filosofi, in cui n filosofi siedono attorno ad una tavola rotonda e fra un filosofo ed un suo vicino si trova un bastoncino (*chopstick*). Per mangiare, un filosofo deve avere i due *chopstick* che si trovano rispettivamente alla sua sinistra e alla sua destra. Implementare

- la classe **Chopstick** con un metodo **pickup** e un metodo **putdown**,
- e la classe **Filosofo** che esegue un ciclo in cui alterna una fase in cui pensa ad una in cui mangia.