SISTEMI OPERATIVI IIN/IEL/IDT
INFORMATICA INDUSTRIALE E SISTEMI OPERATIVI IDI
SISTEMI DI ELABORAZIONE P.O.
prova scritta del 22.06.2005

Nome: _____ Cognome: _____

Sia data una classe Sottoscrittore che definisce un thread come descritto nel seguito:

```
public class Utente extends Thread {
    private Erogatore e = null;

    public Utente ( Erogatore e ) { this.e = e; }

    /*    ...   completare  se necessario     ...   */

    public void run() {
          /*    ...   completare  se necessario     ...   */

          int i = 0;
          while ( true ) {
                i++;
                Object o = e.richiedi( i );
          }
    }
}
```

Come è possibile evincere dal codice riportato sopra, l'utente chiede ad un erogatore oggetti con un identificativo specificato, sempre crescente.

Tenendo presente che sul sistema in esame sono attivi N thread di tipo Utente, si realizzi la classe Erogatore in maniera tale che:

- eroghi l'oggetto *i*-esimo a tutti gli utenti che ne fanno richiesta;
- prima di procedere all'erogazione dell'oggetto *(i+1)*-esimo, deve aver erogato l'oggetto *i*-esimo a tutti gli utenti.
- gli oggetti creati dall'erogatore siano accodati per la distribuzione;
- gli utenti che richiedono un oggetto che non è ancora in erogazione devono essere sospesi.

Soluzione

```
public class Erogatore {
    private int id_erogazione; // id dell'oggetto attualmente in erogazione
    private int count;         // numero di utenti da servire per un dato id
    private int N;             // numero complessivo di utenti
    private Vector oggetti;

    public Erogatore (int numero_utenti) {
        N = numero_utenti;
        id_erogazione = 1;      // dal codice del metodo run() della classe
                                // Utente si osserva che il primo elemento
                                // erogato ha indice 1
        count = numero_utenti;
        oggetti = new Vector(); // si suppone che il vettore sia inizializzato con
                                // gli oggetti ed abbia dimensione indefinita
    }

    public synchronized Object richiedi (int id_richiesta) {
        if (id_richiesta != id_erogazione) {
            try {
                wait();
            }
            catch (InterruptedException e) {
            }
        }
        count --;
        if (count==0) {
            count = N;
            id_erogazione ++;
            notifyAll();

            // gli oggetti nel vettore iniziano da indice 0
            return oggetti.elementAt(id_erogazione-2);
        }
        return oggetti.elementAt(id_erogazione-1);
    }

}


public class Principale {

    public static void main (String args[]) {
        int N = 20;

        Erogatore e = new Erogatore(N);
        for (int i=0;i<N;i++) {
            Utente u = new Utente(e);
            u.start();
        }
    }

}
```