Corso di Programmazione in Rete e Laboratorio
prova scritta
a.a. 2003/2004 – 23 marzo 20034

1. Sia dato il seguente frammento di codice Java:

```
abstract class ContoCorrente {
    private float tasso;
    private float saldo;
    public ContoCorrente(float ts) {
        tasso = ts;
    }
    public float getCompetenze() {
        return getInteressi() - getSpese();
    }
    [...]
}

class CCNumOperazioni extends ContoCorrente {
    private int numOperazioni;
    private float costoOperazione;
    public CCNumOperazioni(float tasso, float co) {
        super(tasso);
        costoOperazione = co;
    }
    [...]
}

class CCSpeseForfettarie extends ContoCorrente {
    private float speseForfettarie;
    public CCSpeseForfettarie (float tasso, float sf) {
        super(tasso);
        speseForfettarie = sf;
    }
    [...]
}
```

a) Lo si completi definendo nella maniera più opportuna i metodi **getInteressi** e **getSpese** in modo da poter calcolare le competenze a fine anno su un insieme di conti correnti sia di tipo **CCSpeseForfettarie** e **CCNumOperazioni** invocando su di essi indistintamente il metodo **getCompetenze**. Si assuma che gli interessi vengano calcolati mediante la formula "saldo * tasso / 100", mentre le spese sono calcolate in maniera forfettaria per i conti con spesa forfettaria (**CCSpeseForfettarie**), con la formula "numero operazioni * costo operazioni" per i conti con numero di operazioni (**CCNumOperazioni**).

b) Si estenda ulteriormente il codice del precedente esercizio in modo da contare il numero totale di conti correnti creati, il numero totale di conti correnti con spese forfettarie creati e il numero totale di conti correnti con numero di operazioni creati, introducendo anche i metodi **getNumTotCC**, **getNumTotCCSF** e **getNumTotCCNO**

che restituiscono i suddetti numeri.

c) Si modifichi la definizione del metodo **getCompetenze** in **ContoCorrente** in modo tale che lanci un'eccezione di **CompetenzeNegative** se le spese sono maggiori degli interessi maturati. Si definisca anche l'*eccezione* **CompetenzeNegative** in modo che contenga il conto che l'ha generata.

2. Sia data la seguente classe

```
public class CubbyHole {
   private int contents;
   private boolean available = false;

   public synchronized int get() {
      while (available == false) {
         try {
              wait();
         } catch (InterruptedException e) { }
      }
      available = false;
      notifyAll();
      return contents;
   }
   public synchronized void put(int value) {
      while (available == true) {
         try {
              wait();
         } catch (InterruptedException e) { }
      }
      contents = value;
      available = true;
      notifyAll();
    }
}
```

a) Dato un oggetto c della classe **CubbyHole**, si assuma che ci siano tre thread distinti t1, t2 e t3 che eseguono rispettivamente i metodi c.get() c.get() e c.put(10), e si assuma inoltre che le tre chiamate vengano attivate dallo scheduler nell'ordine dato. Descrivere in dettaglio i passi dell'esecuzione, con particolare riferimento alle operazioni di wait e notifyAll.

b) Cambierebbe qualcosa se nei metodi get e put ci fosse un **if** al posto del **while**?

3. Definire una sottoclasse di **JFrame** che realizzi una finestra contenente una **JLabel**, inizializzata a 0, e due **JButton** tali che, quando si preme il primo il contenuto della **JLabel** venga incrementato di 1 e quando si preme il secondo venga decrementato di 1.