



Università degli Studi dell'Insubria
Dipartimento di Scienze Teoriche e Applicate

Programmazione Concorrente e Distribuita Client-server con socket

Luigi Lavazza
Dipartimento di Scienze Teoriche e Applicate
luigi.lavazza@uninsubria.it



Esercizio 1

- Si modifichi il codice dato, per creare un sistema client-server, in cui client e server comunicano attraverso socket.



class Operation

```
public class Operation {  
    int CCnumber;  
    int amount;  
    String type;  
    Operation(int cc, int val, String what){  
        CCnumber=cc;  
        amount=val;  
        type=what;  
    }  
    public int getCCnumber() {  
        return CCnumber;  
    }  
    public int getAmount() {  
        return amount;  
    }  
    public String getType() {  
        return type;  
    }  
}
```



class Client

```
public class Client {
    private void printResult(Result r) {
        if(r.isSuccessful()) {
            System.out.println("success; total is "+r.getAmount());
        } else {
            System.out.println("operation failed");
        }
    }
    public void start(Bank b) {
        Result r;
        r=b.executeOperation(new Operation(1, 55, "Deposit"));
        printResult(r);
        r=b.executeOperation(new Operation(1, 82, "Withdraw"));
        printResult(r);
        r=b.executeOperation(new Operation(1, 22, "Withdraw"));
        printResult(r);
    }
    public static void main(String[] args) {
        Bank myBank = new Bank();
        Client c=new Client();
        c.start(myBank);
    }
}
```



class Bank

```
public class Bank {
    int ccAmounts[]={0,0,0};
    public Result executeOperation(Operation op) {
        int ccNum=op.getCCnumber();
        if(ccNum<0 || ccNum>2) { return new Result(-1, 0, false); }
        String opType=op.getType();
        if(opType.equals("Deposit")) {
            ccAmounts[ccNum]+=op.getAmount();
            return new Result(ccNum, ccAmounts[ccNum], true);
        } else if (opType.equals("Withdraw")) {
            if(op.getAmount()>ccAmounts[ccNum]) {
                return new Result(-1, 0, false);
            } else {
                ccAmounts[ccNum]-=op.getAmount();
                return new Result(ccNum, ccAmounts[ccNum], true);
            }
        } else {
            return new Result(-1, 0, false);
        }
    }
}
```



class Result

```
public class Result {
    int CCnumber;
    int amount;
    boolean successful;
    Result(int cc, int val, boolean ok){
        CCnumber=cc;
        amount=val;
        successful=ok;
    }
    public int getCCnumber() {
        return CCnumber;
    }
    public int getAmount() {
        return amount;
    }
    public boolean isSuccessful() {
        return successful;
    }
}
```



Esercizio 2

- Si modifichi il programma sviluppato nell'esercizio 1 per tenere conto dei requisiti seguenti:
 - ▶ L'operazione richiesta può richiedere parecchio tempo.
 - ▶ Il client, una volta mandata la richiesta alla banca, non resta in attesa della risposta, ma continua ad eseguire le proprie elaborazioni.
 - ▶ Quando l'operazione è completata, la banca manda un avviso al client.
 - L'avviso comprende il valore del saldo del CC del client.
 - ▶ Quando il client riceve l'avviso e legge il valore del saldo, visualizza l'esito dell'operazione.