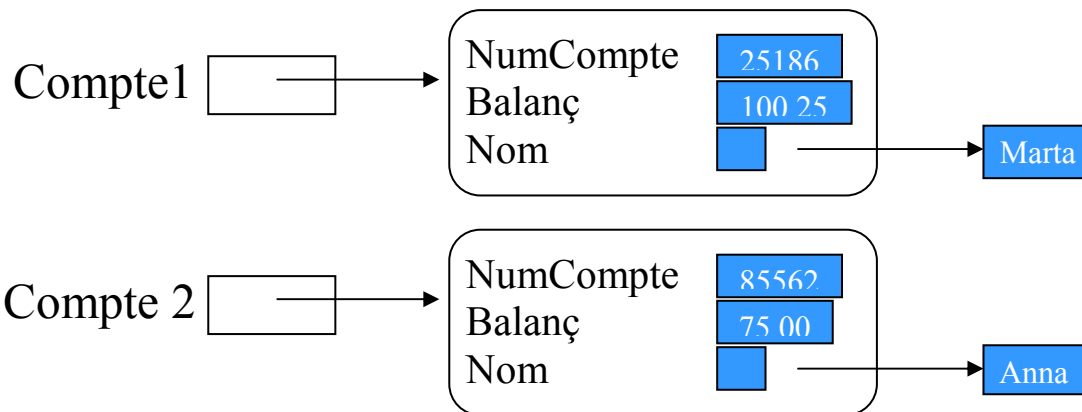


Metodologia i Tecnologia de la Programació

Exercici 1 – Implementeu una classe que :

- Representi un compte bancari mitjançant una classe anomenada Account
- El seu estat inclou el numero de compte, el saldo actual i el nom del propietari
- Els serveis són: afegir o extraure diners, afegir interessos i obtenir un balanç dels diners



Solució:

```

//*****
// Account.java
//*****

import java.text.NumberFormat;

public class Account
{
    private final double RATE = 0.035; // constant for interest rate of 3.5%

    private long acctNumber;
    private double balance;
    private String name;

    //-----
    // Sets up the account by defining its owner, account number,
    // and initial balance.
    //-----
    public Account (String owner, long account, double initial)
    {
        name = owner;
        acctNumber = account;
        balance = initial;
    }

    //-----
    // Deposits the specified amount into the account. Returns the
    // new balance.
    //-----
    public double deposit (double amount)
    {
        balance = balance + amount;

        return balance;
    }

    //-----
    // Withdraws the specified amount from the account and applies

```

```

// the fee. Returns the new balance.
//-----
public double withdraw (double amount, double fee)
{
    balance = balance - amount - fee;

    return balance;
}

//-----
// Adds interest to the account and returns the new balance.
//-----
public double addInterest ()
{
    balance += (balance * RATE);
    return balance;
}

//-----
// Returns the current balance of the account.
//-----
public double getBalance ()
{
    return balance;
}

//-----
// Returns a one-line description of the account as a string.
//-----
public String toString ()
{
    NumberFormat fmt = NumberFormat.getCurrencyInstance();

    return (acctNumber + "\t" + name + "\t" + fmt.format(balance));
}
}

```

Exercici 2 – Implementeu una classe *Transactions* que :

- Utilitzi la classe *Account* i que declari diferents comptes
- Afegeixi diners a cada compte
- Permeti retirar diners d'un compte
- Comprovar el balanç d'algun dels comptes

Solució:

```

//*****
// Transactions.java
//*****

public class Transactions
{
    //-----
    // Creates some bank accounts and requests various services.
    //-----
    public static void main (String[] args)
    {
        Account acct1 = new Account ("Ted Murphy", 72354, 102.56);
        Account acct2 = new Account ("Jane Smith", 69713, 40.00);
        Account acct3 = new Account ("Edward Demsey", 93757, 759.32);

        acct1.deposit (25.85);

        double smithBalance = acct2.deposit (500.00);
        System.out.println ("Smith balance after deposit: " + smithBalance);

        System.out.println ("Murphy balance after withdrawal: " + acct2.withdraw (430.75, 1.50));

        acct1.addInterest();
        acct2.addInterest();
        acct3.addInterest();
    }
}

```

```

        System.out.println ();
        System.out.println (acct1);
        System.out.println (acct2);
        System.out.println (acct3);
    }
}

```

Exercici 3 – Implementeu una classe *Vacances* on :

- El programa sol·licita el dia, mes i any de sortida del viatge i la durada del mateix. Si la data és superior al dia d'avui, mostra la data de final del viatge, sinó la data d'inici.
- Consulteu el javadoc de les classes *java.util.Date* i *java.util.GregorianCalendar* i utilitzeu el que creieu oportú.

Solució:

```

//*****
//  Vacances.java
//*****

import java.util.Date;
import java.util.GregorianCalendar;
import java.util.Calendar;

public class Vacances {
    private Date dataInici;
    private Date dataFinal;
    private int durada;
    private GregorianCalendar calendar;

    public Vacances(int dia, int mes, int any){
        durada = 0;
        calendar = new GregorianCalendar();
        calendar.set(any, mes, dia);
        dataInici = calendar.getTime();
        dataFinal = calendar.getTime();
    }

    public void setDies(int dies){
        durada = dies;
    }
    public Date getInici(){
        return dataInici;
    }
    public Date getFinal(){

        Date avui = new Date();
        if (avui.before(dataInici)) {
            calendar.add(Calendar.DATE, durada);
            dataFinal = (Date) calendar.getTime();
        }

        return dataFinal;
    }

    public static void main(String[] args){
        if (args.length < 4) {
            System.out.println ("Falten params: vacances <dia> <mes> <any> <durada> ");
        }
        else {
            int dia = Integer.parseInt(args[0]);
            int mes = Integer.parseInt(args[1]); // 0 gener, 1 febrer, etc.
            int any = Integer.parseInt(args[2]);

            int durada = Integer.parseInt(args[3]);

            Vacances vac = new Vacances(dia, mes, any);
            vac.setDies(durada);
            System.out.println("Les vacances comencen: " + vac.getInici());
            System.out.println("I acaben " + vac.getFinal());
        }
    }
}

```