

Examen Parcial (Solució)

Maig de 2014

Nom i Cognoms:

NIUB:

Exercici 1 (8 punts)

Se'ns ha encomanat dissenyar una eina adreçada als empleats d'una empresa, que permeti entre altres funcionalitats, calcular el seu sou segons la informació emmagatzemada de cadascun d'ells. En concret, cal desenvolupar una interfície gràfica d'usuari que permeti entrar les dades de cada empleat i calcular amb aquestes el seu sou mensual.

En la primera versió de la interfície els empleats seran o Administratius o Tècnics Informàtics. De cada empleat caldrà guardar el seu nom, cognoms, el seu DNI i el seu sou base. A més, en el cas del tècnic informàtic, es necessita addicionalment informació del número d'hores extra realitzades. Pel càlcul del sou final, en el cas dels administratius, el sou final serà el sou base i en el cas dels tècnics informàtics serà igual al sou base més el número d'hores extres multiplicat per 20€ cadascuna.

L'aplicació constarà d'un botó per crear un nou empleat i un botó per calcular el sou final d'un empleat.

Les classes requerides seran:

1. Aplicacio
2. BotoNouEmpleatListener
3. BotoSouFinalListener
4. BotoPressupostSousListener
5. Empleat
6. Administratiu
7. TècnicInformatic

Les classes BotoNouUsuariListener, BotoSouFinalListener i BotoPressupostSousListener són les classes de tipus Listeners pels diferents botons de l'aplicació. Les classes Empleat, Administratiu, TècnicInformatic són les classes que guarden les dades dels empleats de l'aplicació.

A continuació presentem la implementació parcial de la classe Aplicacio i de la classe abstracta Empleat. Llegiu-les atentament i realitzeu els exercicis indicats en fulles en blanc.

Aplicacio.java

1. package examen;
2. import java.awt.BorderLayout;
3. import java.awt.event.ActionEvent;
4. import java.awt.event.ActionListener;
5. import java.util.ArrayList;
6. import java.util.Iterator;
7. import javax.swing.BoxLayout;
8. import javax.swing.JButton;
9. import javax.swing.JFrame;
10. import javax.swing.JLabel;
11. import javax.swing.JPanel;
12. import javax.swing.JTextField;
13. public class Aplicacio{
14. JFrame frame;

Programació 2.

Grau d'Enginyeria Informàtica. Facultat de Matemàtiques. UB
Curs 2013-2014

```
15.      JTextField txtInfo;
16.      int numEmpleatActual;
17.      ArrayList<Empleat> llistaEmpleats;
18.      JTextField txtPressupostTotal;
19.
20.      public Aplicacio(){
21.          frame = new JFrame("Aplicació de gestió d'Empleats");
22.          txtInfo = new JTextField("");
23.          numEmpleatActual=0;
24.          llistaEmpleats = new ArrayList<Empleat>();
25.          txtPressupostTotal = new JTextField("");
26.      }
27.
28.      public void go(){
29.          JButton btnNouEmpleat;
30.          JButton btnSouFinal;
31.          JButton btnPressupostTotal;
32.          JPanel panel = new JPanel();
33.          JLabel labelInfo = new JLabel("Informació Empleat:");
34.          JLabel labelPressupostTotal = new JLabel("Pressupost Total:");
35.
36.          btnNouEmpleat = new JButton("Crea nou empleat");
37.          btnSouFinal = new JButton("Calcula Sou Final");
38.          btnPressupostTotal = new JButton("Calcula Pressupost Total");
39.
40.          frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
41.          panel.setLayout(new BoxLayout(panel,BoxLayout.Y_AXIS));
42.
43.          panel.add(btnNouEmpleat);
44.          panel.add(btnSouFinal);
45.          panel.add(labelInfo);
46.          panel.add(txtInfo);
47.          panel.add(btnPressupostTotal);
48.          panel.add(labelPressupostTotal);
49.          panel.add(txtPressupostTotal);
50.          frame.getContentPane().add(BorderLayout.NORTH, panel);
51.          frame.setSize(400,400);
52.          frame.setVisible(true);
53.      } // Final mètode go
54.
55.      public static void main(String [] args){
56.          Aplicacio aplicacio = new Aplicacio();
57.          aplicacio.go();
58.      }
59.  } // Final classe Aplicacio
```

Empleat.java

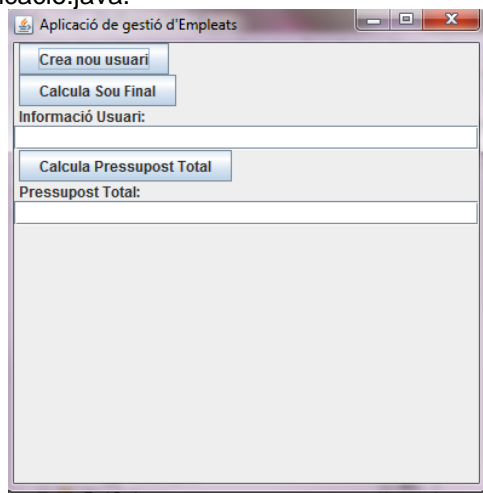
```
1.      package examen;
2.
3.      public abstract class Empleat {
4.          private String nom;
5.          private int dni;
6.          private float souBase;
7.
8.          public Empleat (String nom, int dni, float souBase){
9.              this.nom = nom;
10.             this.dni = dni;
11.             this.souBase = souBase;
```

Programació 2.

Grau d'Enginyeria Informàtica. Facultat de Matemàtiques. UB
Curs 2013-2014

```
12.     }
13.     public abstract float calculaSouFinal();
14.
15.     public void setNom(String nom){
16.         this.nom = nom;
17.     }
18.     public String getNom(){
19.         return nom;
20.     }
21.     public void setDNI(int dni){
22.         this.dni = dni;
23.     }
24.     public int getDNI(){
25.         return dni;
26.     }
27.     public void setSouBase(float sou){
28.         this.souBase = sou;
29.     }
30.     public float getSouBase(){
31.         return souBase;
32.     }
33. }
```

- a) **(1 punt)** Feu un dibuix de com creus que quedarà la interfície gràfica d'usuari definida a Aplicacio.java.



- b) **(0.5 punts)** Expliqueu què s'aconsegueix fent servir la següent assignació:
`panel.setLayout(new BorderLayout(panel, BorderLayout.Y_AXIS));`

Aconsegüim introduir les components en vertical

- c) **(0.5 punt)** Indica quin és el manejador de components (layout manager) per defecte de la classe JFrame en Java i en què consisteix.

El manejador de components (layout manager) per defecte de la classe JFrame en Java és el **BorderLayout** que divideix l'espai en 5 parts (nord, sud, est, oest i centre) i introdueix una component en cada part.

- d) **(1.5 punts)** Implementeu la classe 'TecniciInformatic' i 'Administratiu' que hereten de la classe Empleat. No cal que implementeu els consultors i modificadors.

Programació 2.

Grau d'Enginyeria Informàtica. Facultat de Matemàtiques. UB
Curs 2013-2014

Solució:

TecnicInformatic.java

```
package examen;

public class TecnicInformatic extends Empleat{
    int horesExtres;

    public TecnicInformatic(String nom, int dni, float souBase, int horesExtres){
        super(nom, dni, souBase);
        this.horesExtres = horesExtres;
    }
    public float calculaSouFinal() {
        float sou;
        sou = super.getSouBase() + horesExtres*20;
        return sou;
    }
}
```

Administratiu.java

```
package examen;

public class Administratiu extends Empleat{

    public Administratiu(String nom, int dni, float souBase){
        super(nom, dni, souBase);
    }
    public float calculaSouFinal() {
        float sou;
        sou = super.getSouBase();
        return sou;
    }
}
```

- e) **(1 punts)** Indiqueu quin és el codi necessari que s'hauria d'afegir al codi presentat a l'enunciat per tal que quan es prem el botó 'btnSouFinal' de l'aplicació:
- Es calculi el sou final de l'empleat actual, corresponent a l'empleat amb posició numEmpleatActual (guardat com atribut).
 - Es mostri el valor resultant del càlcul al camp de text txtInfo.

Tingueu en compte la llista de noms de classes donada a l'enunciat. Especifiqueu en quines línies del codi donat afegiu cada part de codi nou.

Programació 2.

Grau d'Enginyeria Informàtica. Facultat de Matemàtiques. UB
Curs 2013-2014

Solució:

A la línia 42:

```
btnSouFinal.addActionListener(new BotoSouFinalListener());
```

A la línia 54:

```
class BotoSouFinalListener implements ActionListener {  
    public void actionPerformed(ActionEvent event){  
        float f = llistaEmpleats.get(numEmpleatActual - 1).calculaSouFinal();  
        txtInfo.setText(Float.toString(f));  
    }  
}
```

- f) **(0.5 punts)** Indiqueu quins objectes estaran involucrats en aquesta operació, segons el model de gestió d'events. Fes un esquema per explicar-ho.

En aquest cas, els objectes involucrats en el model de gestió d'events són l'objecte escoltador de tipus **BotoSouFinalListener**, l'objecte event de tipus **ActionEvent** i l'objecte component de tipus **JButton**.

- g) **(0.5 punts)** Expliqueu quines diferències hi ha si es declara la classe BotoSouFinalListener com a classe externa de la classe Aplicacio.

Solució:

Si aquesta classe es declara com a classe interna podrà accedir als atributs de la classe principal (Aplicacio) sense necessitat de passar-li l'objecte com a paràmetre. Si es declara com a externa haurà de tenir l'objecte de tipus Aplicacio com a paràmetre per tal de poder actualitzar el camp de text corresponent.

- h) **(1 punts)** Volem afegir una nova funcionalitat a l'aplicació: quan es prem el botó "btnPressupostTotal", volem que es calculi el pressupost total, es a dir, la suma dels sous finals de cada empleat segons la llista d'empleats existent en aquell moment. Assumiu que aquesta llista ha estat emplenada previament.

Implementeu una classe BotoPressupostSousListener que faci aquest càlcul i el mostri al camp de text txtPressupostTotal.

Indiqueu quin és el codi necessari que s'hauria d'afegir al codi presentat a l'enunciat i especifiqueu en quines línies del codi donat afegiu cada part de codi nou.

A la línia 54:

```
class BotoPressupostSousListener implements ActionListener {  
    public void actionPerformed(ActionEvent event){  
        float pressupost=0;  
        if(llistaEmpleats!=null) {  
            for(Iterator<Empleat> i = llistaEmpleats.iterator(); i.hasNext();) {  
                pressupost = pressupost + i.next().calculaSouFinal();  
            }  
        }  
        String strPressupost = Float.toString(pressupost);  
        txtPressupostTotal.setText(strPressupost);  
    }  
}
```

Programació 2.

Grau d'Enginyeria Informàtica. Facultat de Matemàtiques. UB
Curs 2013-2014

```
}
```

A la línia 42:

```
btnPressupostTotal.addActionListener(new BotoPressupostSousListener());
```

- i) **(1 punts)** Canvia el codi de l'exercici anterior per fer servir una classe interna anònima.

A la línia 42:

```
btnPressupostTotal.addActionListener(new ActionListener(){  
    public void actionPerformed(ActionEvent event){  
        float pressupost=0;  
        if(llistaEmpleats!=null) {  
            for(Iterator<Empleat> i = llistaEmpleats.iterator(); i.hasNext();) {  
                pressupost = pressupost + i.next().calculaSouFinal();  
            }  
        }  
        String strPressupost = Float.toString(pressupost);  
        txtPressupostTotal.setText(strPressupost);  
    });
```

- j) **(0.5 punts)** Indiqueu quin valor apareixerà al camp de text del pressupost total si a la llista d'empleats hem afegit els següents empleats:

```
Empleat empleat1 = new TecnicInformatic("Maria", 111111, 1000, 5);  
llistaEmpleats.add(empleat1);  
Empleat empleat2 = new Administratiu("Joan", 111111, 1000);  
llistaEmpleats.add(empleat2);
```

Apareix: 2100.0