

## **Estructuras de secuencia**

---

### **LOGRO DE LA UNIDAD DE APRENDIZAJE**

Al finalizar la unidad, el alumno, empleando las etapas de desarrollo de un algoritmo y las instrucciones algorítmicas básicas, diseña algoritmos secuenciales en pseudocódigo.

### **TEMARIO**

1. Estructuras de secuencia

### **ACTIVIDADES**

Los alumnos desarrollan algoritmos que involucren estructuras de secuencia.

## 1. Estructura de secuencia

Una *estructura de secuencia* es aquella en la que las instrucciones están una a continuación de la otra siguiendo una secuencia única.

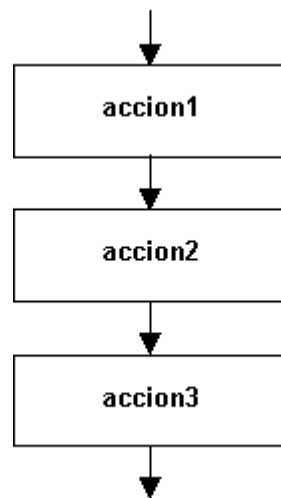


Figura 1 Estructura de Secuencia

## 2. Problemas resueltos

### Problema 1

Diseñe un programa que determine el área total ( $A$ ) y el volumen ( $V$ ) de un cilindro del que se conoce su radio ( $r$ ) y su altura ( $h$ ). Considere las siguientes fórmulas:

$$A = 2\pi r(r + h)$$
$$V = \pi r^2 h$$

### Algoritmo

#### Inicio

```
// Declaración de variables  
real r, h, are, vol
```

```
// Entrada de datos  
Leer r, h
```

```
// Proceso de cálculo  
are = 2*3.1416*r*(r+h)  
vol = 2*3.1416*r*r*h
```

```
// Salida de resultados  
Imprimir are, vol
```

#### Fin

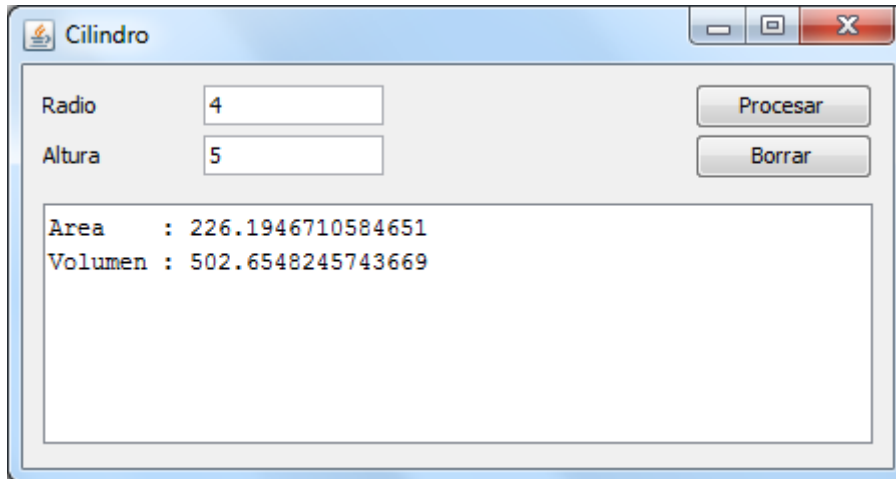
### Programa

```
package cibertec;  
  
import java.awt.EventQueue;  
import java.awt.Font;
```

```

import javax.swing.JFrame;
import javax.swing.JButton;
import javax.swing.UIManager;
import javax.swing.JLabel;
import javax.swing.JTextField;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;

```



```

public class Cilindro extends JFrame implements ActionListener {
    private static final long serialVersionUID = 1L;

    // Declaración de variables
    private JLabel lblRadio;
    private JLabel lblAltura;
    private JTextField txtRadio;
    private JTextField txtAltura;
    private JButton btnProcesar;
    private JButton btnBorrar;
    private JScrollPane scpScroll;
    private JTextArea txtS;

    // Lanza la aplicación
    public static void main(String[] args) {
        try {
            UIManager.setLookAndFeel(
                "com.sun.java.swing.plaf.windows.WindowsLookAndFeel");
        }
        catch (Throwable e) {
            e.printStackTrace();
        }
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    Cilindro frame = new Cilindro();
                    frame.setVisible(true);
                }
                catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }
}

```

```

}

// Crea la GUI
public Cilindro() {
    setTitle("Cilindro");
    setBounds(100, 100, 450, 239);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    getContentPane().setLayout(null);

    lblRadio = new JLabel("Radio");
    lblRadio.setBounds(10, 13, 80, 14);
    getContentPane().add(lblRadio);

    lblAltura = new JLabel("Altura");
    lblAltura.setBounds(10, 38, 80, 14);
    getContentPane().add(lblAltura);

    txtRadio = new JTextField();
    txtRadio.setBounds(90, 10, 90, 20);
    getContentPane().add(txtRadio);
    txtRadio.setColumns(10);

    txtAltura = new JTextField();
    txtAltura.setBounds(90, 35, 90, 20);
    getContentPane().add(txtAltura);
    txtAltura.setColumns(10);

    btnProcesar = new JButton("Procesar");
    btnProcesar.addActionListener(this);
    btnProcesar.setBounds(335, 9, 89, 23);
    getContentPane().add(btnProcesar);

    btnBorrar = new JButton("Borrar");
    btnBorrar.addActionListener(this);
    btnBorrar.setBounds(335, 34, 89, 23);
    getContentPane().add(btnBorrar);

    scpScroll = new JScrollPane();
    scpScroll.setBounds(10, 69, 414, 120);
    getContentPane().add(scpScroll);

    txtS = new JTextArea();
    txtS.setFont(new Font("Monospaced", Font.PLAIN, 12));
    scpScroll.setViewportView(txtS);
}

// Direcciona eventos de tipo ActionEvent
public void actionPerformed(ActionEvent arg0) {
    if (arg0.getSource() == btnProcesar) {
        actionPerformedBtnProcesar(arg0);
    }
    if (arg0.getSource() == btnBorrar) {
        actionPerformedBtnBorrar(arg0);
    }
}

// Procesa la pulsación del botón Procesar
protected void actionPerformedBtnProcesar(ActionEvent arg0) {
    // Declaración de variables

```

```

    double r, h, are, vol;

    // Entrada de datos
    r = Double.parseDouble(txtRadio.getText());
    h = Double.parseDouble(txtAltura.getText());

    // Proceso
    are = 2 * Math.PI * r * (r + h);
    vol = 2 * Math.PI * r * r * h;

    // Salida de resultados
    txtS.setText("Area      : " + are + "\n");
    txtS.append("Volumen : " + vol);
}

// Procesa la pulsación del botón Borrar
protected void actionPerformedBtnBorrar(ActionEvent arg0) {
    txtRadio.setText("");
    txtAltura.setText("");
    txtS.setText("");
    txtRadio.requestFocus();
}
}

```

## Problema 2

Diseñe un programa que determine el área de la base ( $A$ ) y el volumen ( $V$ ) de una pirámide de base rectangular conociendo el largo ( $m$ ) y el ancho ( $n$ ) de la base y la altura ( $h$ ) de la pirámide. Considere las siguientes fórmulas:

$$A = m n$$

$$V = \frac{A h}{3}$$

## Algoritmo

### Inicio

```

// Declaración de variables
real m, n, h, are, vol

```

```

// Entrada de datos
Leer m, n, h

```

```

// Proceso de cálculo
are = m * n
vol = are * h / 3

```

```

// Salida de resultados
Imprimir are, vol

```

Fin

## Programa

```

package cibertec;

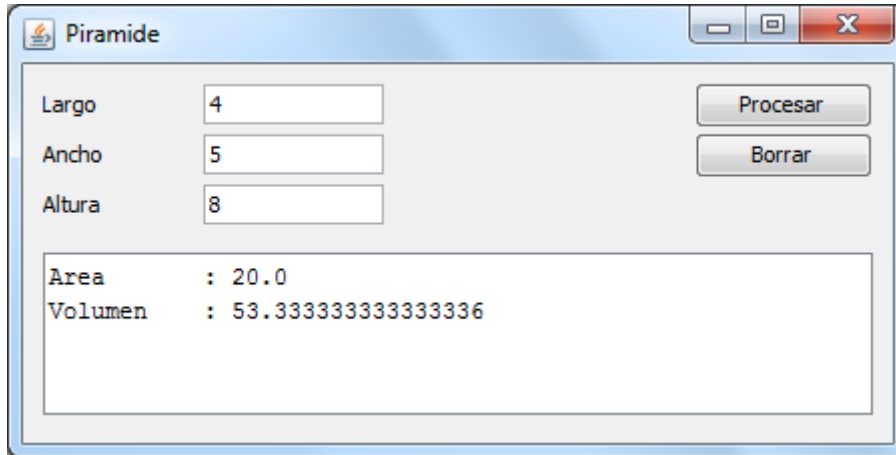
import java.awt.EventQueue;
import java.awt.Font;
import javax.swing.JFrame;

```

```

import javax.swing.JButton;
import javax.swing.UIManager;
import javax.swing.JLabel;
import javax.swing.JTextField;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;

```



```

public class Piramide extends JFrame implements ActionListener {
    private static final long serialVersionUID = 1L;

    // Declaración de variables
    private JLabel lblLargo;
    private JLabel lblAncho;
    private JLabel lblAltura;
    private JTextField txtLargo;
    private JTextField txtAncho;
    private JTextField txtAltura;
    private JButton btnProcesar;
    private JButton btnBorrar;
    private JScrollPane scpScroll;
    private JTextArea txtS;

    // Lanza la aplicación
    public static void main(String[] args) {
        try {
            UIManager.setLookAndFeel(
                "com.sun.java.swing.plaf.windows.WindowsLookAndFeel");
        }
        catch (Throwable e) {
            e.printStackTrace();
        }
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    Piramide frame = new Piramide();
                    frame.setVisible(true);
                }
                catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }
}

```

```

}

// Crea la GUI
public Piramide() {
    setTitle("Piramide");
    setBounds(100, 100, 450, 227);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    getContentPane().setLayout(null);

    lblLargo = new JLabel("Largo");
    lblLargo.setBounds(10, 13, 80, 14);
    getContentPane().add(lblLargo);

    lblAncho = new JLabel("Ancho");
    lblAncho.setBounds(10, 38, 80, 14);
    getContentPane().add(lblAncho);

    lblAltura = new JLabel("Altura");
    lblAltura.setBounds(10, 63, 80, 14);
    getContentPane().add(lblAltura);

    txtLargo = new JTextField();
    txtLargo.setBounds(90, 10, 90, 20);
    getContentPane().add(txtLargo);
    txtLargo.setColumns(10);

    txtAncho = new JTextField();
    txtAncho.setBounds(90, 35, 90, 20);
    getContentPane().add(txtAncho);
    txtAncho.setColumns(10);

    txtAltura = new JTextField();
    txtAltura.setBounds(90, 60, 90, 20);
    getContentPane().add(txtAltura);
    txtAltura.setColumns(10);

    btnProcesar = new JButton("Procesar");
    btnProcesar.addActionListener(this);
    btnProcesar.setBounds(335, 9, 89, 23);
    getContentPane().add(btnProcesar);

    btnBorrar = new JButton("Borrar");
    btnBorrar.addActionListener(this);
    btnBorrar.setBounds(335, 34, 89, 23);
    getContentPane().add(btnBorrar);

    scpScroll = new JScrollPane();
    scpScroll.setBounds(10, 94, 414, 81);
    getContentPane().add(scpScroll);

    txtS = new JTextArea();
    txtS.setFont(new Font("Monospaced", Font.PLAIN, 12));
    scpScroll.setViewportView(txtS);
}

// Direcciona eventos de tipo ActionEvent
public void actionPerformed(ActionEvent arg0) {
    if (arg0.getSource() == btnProcesar) {
        actionPerformedBtnProcesar(arg0);
    }
}

```

```

    }
    if (arg0.getSource() == btnBorrar) {
        actionPerformedBtnBorrar(arg0);
    }
}

// Procesa la pulsación del botón Procesar
protected void actionPerformedBtnProcesar(ActionEvent arg0) {
    // Declaración de variables
    double m, n, h, are, vol;

    // Entrada de datos
    m = Double.parseDouble(txtLargo.getText());
    n = Double.parseDouble(txtAncho.getText());
    h = Double.parseDouble(txtAltura.getText());

    // Proceso de cálculo
    are = m * n;
    vol = are * h / 3;

    // Salida de resultados
    txtS.setText("Area      : " + are + "\n");
    txtS.append("Volumen    : " + vol + "\n");
}

// Procesa la pulsación del botón Borrar
protected void actionPerformedBtnBorrar(ActionEvent arg0) {
    txtLargo.setText("");
    txtAncho.setText("");
    txtAltura.setText("");
    txtS.setText("");
    txtLargo.requestFocus();
}
}

```

### Problema 3

Una empresa expondrá sus productos en una feria. La empresa considera que el monto total de dinero a invertir estará distribuido de la siguiente manera:

| Rubro                           | Porcentaje |
|---------------------------------|------------|
| Alquiler de espacio en la feria | 23%        |
| Publicidad                      | 7%         |
| Transporte                      | 26%        |
| Servicios feriales              | 12%        |
| Decoración                      | 21%        |
| Gastos varios                   | 11%        |

Dado el monto total de dinero a invertir, diseñe un programa que determine cuánto gastará la empresa en cada rubro.

### Algoritmo

#### Inicio

```

// Declaración de variables
real montoTotal, rubro1, rubro2, rubro3, rubro4, rubro5, rubro6

```



```

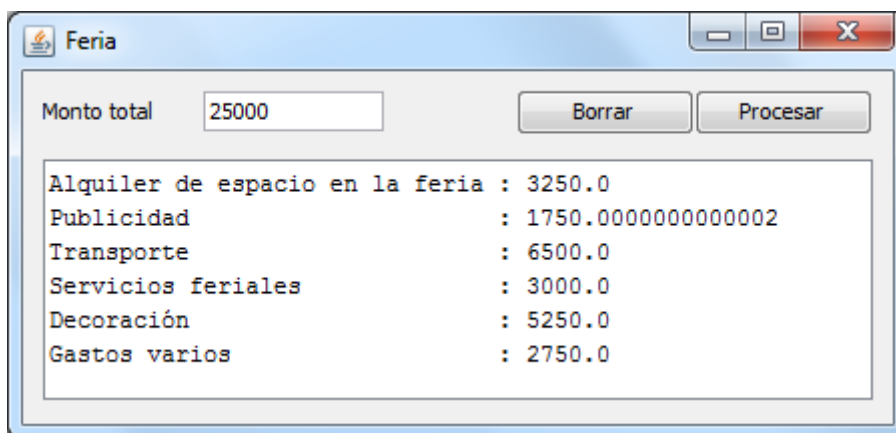
// Entrada de datos
Leer montoTotal

// Proceso de cálculo
rubro1 = 0.13*montoTotal
rubro2 = 0.07*montoTotal
rubro3 = 0.26*montoTotal
rubro4 = 0.12*montoTotal
rubro5 = 0.21*montoTotal
rubro6 = 0.11*montoTotal

// Salida de resultados
Imprimir rubro1, rubro2, rubro3, rubro4, rubro5, rubro6
Fin

```

## Programa



```

package cibertec;

import java.awt.EventQueue;
import java.awt.Font;
import javax.swing.JFrame;
import javax.swing.JButton;
import javax.swing.UIManager;
import javax.swing.JLabel;
import javax.swing.JTextField;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;

public class Feria extends JFrame implements ActionListener {
    private static final long serialVersionUID = 1L;

    // Declaración de variables
    private JLabel lblMontoTotal;
    private JTextField txtMontoTotal;
    private JButton btnProcesar;
    private JButton btnBorrar;
    private JScrollPane scpScroll;
    private JTextArea txtS;

    // Lanza la aplicación

```

```

public static void main(String[] args) {
    try {
        UIManager.setLookAndFeel(
            "com.sun.java.swing.plaf.windows.WindowsLookAndFeel");
    }
    catch (Throwable e) {
        e.printStackTrace();
    }
    EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                Feria frame = new Feria();
                frame.setVisible(true);
            }
            catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}

// Crea la GUI
public Feria() {
    setTitle("Feria");
    setBounds(100, 100, 450, 214);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    getContentPane().setLayout(null);

    lblMontoTotal = new JLabel("Monto total");
    lblMontoTotal.setBounds(10, 13, 80, 14);
    getContentPane().add(lblMontoTotal);

    txtMontoTotal = new JTextField();
    txtMontoTotal.setBounds(90, 10, 90, 20);
    getContentPane().add(txtMontoTotal);
    txtMontoTotal.setColumns(10);

    btnProcesar = new JButton("Procesar");
    btnProcesar.addActionListener(this);
    btnProcesar.setBounds(335, 9, 89, 23);
    getContentPane().add(btnProcesar);

    btnBorrar = new JButton("Borrar");
    btnBorrar.addActionListener(this);
    btnBorrar.setBounds(246, 9, 89, 23);
    getContentPane().add(btnBorrar);

    scpScroll = new JScrollPane();
    scpScroll.setBounds(10, 44, 414, 120);
    getContentPane().add(scpScroll);

    txtS = new JTextArea();
    txtS.setFont(new Font("Monospaced", Font.PLAIN, 12));
    scpScroll.setViewportView(txtS);
}

// Direcciona eventos de tipo ActionEvent
public void actionPerformed(ActionEvent arg0) {
    if (arg0.getSource() == btnProcesar) {

```

```

        actionPerformedBtnProcesar(arg0);
    }
    if (arg0.getSource() == btnBorrar) {
        actionPerformedBtnBorrar(arg0);
    }
}

// Procesa la pulsación del botón Procesar
protected void actionPerformedBtnProcesar(ActionEvent arg0) {
    // Declaración de variables
    double montoTotal, rubro1, rubro2, rubro3, rubro4, rubro5, rubro6;

    // Entrada de datos
    montoTotal = Double.parseDouble(txtMontoTotal.getText());

    // Proceso de cálculo
    rubro1 = 0.13*montoTotal;
    rubro2 = 0.07*montoTotal;
    rubro3 = 0.26*montoTotal;
    rubro4 = 0.12*montoTotal;
    rubro5 = 0.21*montoTotal;
    rubro6 = 0.11*montoTotal;

    // Salida de resultados
    txtS.setText("Alquiler de espacio en la feria : " + rubro1 + "\n");
    txtS.append ("Publicidad : " + rubro2 + "\n");
    txtS.append ("Transporte : " + rubro3 + "\n");
    txtS.append ("Servicios feriales : " + rubro4 + "\n");
    txtS.append ("Decoración : " + rubro5 + "\n");
    txtS.append ("Gastos varios : " + rubro6);
}

// Procesa la pulsación del botón Borrar
protected void actionPerformedBtnBorrar(ActionEvent arg0) {
    txtMontoTotal.setText("");
    txtS.setText("");
    txtMontoTotal.requestFocus();
}
}

```

#### Problema 4

Un padre repartirá una cantidad de dinero entre sus cinco hijos. Cada uno recibirá una cantidad equivalente a:

- Tamar: 85% del monto recibido por Josué
- Josué: 27% de la cantidad a repartir
- Caleb: 23% del monto total recibido entre Josué y Daniel
- Daniel: 25% de la cantidad a repartir
- David: lo que queda del dinero a repartir

Dada la cantidad de dinero a repartir, diseñe un programa que determine cuánto de dinero recibirá cada hijo.

#### Algoritmo

##### Inicio

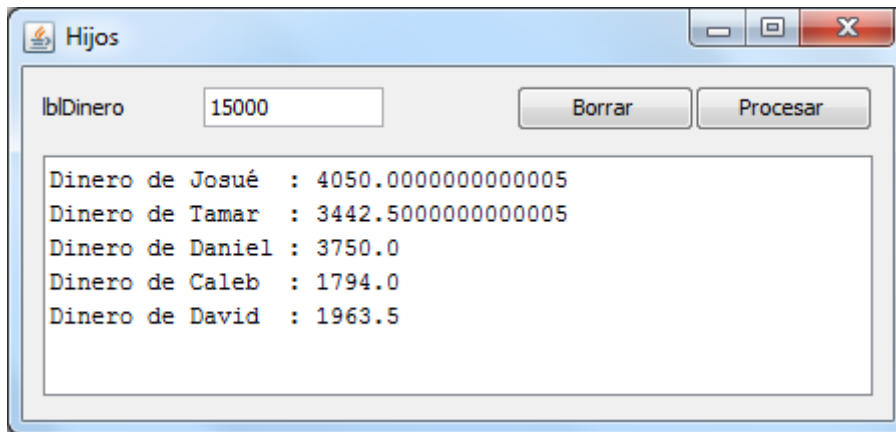
```
// Declaración de variables
real dinero, dinTamar, dinJosue, dinCaleb, dinDaniel, dinDavid

// Entrada de datos
Leer dinero

// Proceso de cálculo
dinJosue = 0.27*dinero
dinTamar = 0.85*dinJosue
dinDaniel = 0.25*dinero
dinCaleb = 0.23*(dinJosue + dinDaniel)
dinDavid = dinero - (dinTamar + dinJosue + dinCaleb + dinDaniel)

// Salida de resultados
Imprimir dinJosue, dinTamar, dinDaniel, dinCaleb, dinDavid
Fin
```

## Programa



```
package cibertec;

import java.awt.EventQueue;
import java.awt.Font;
import javax.swing.JFrame;
import javax.swing.JButton;
import javax.swing.UIManager;
import javax.swing.JLabel;
import javax.swing.JTextField;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;

public class Hijos extends JFrame implements ActionListener {
    private static final long serialVersionUID = 1L;

    // Declaración de variables
    private JLabel Dinero;
    private JTextField txtDinero;
    private JButton btnProcesar;
    private JButton btnBorrar;
    private JScrollPane scpScroll;
    private JTextArea txtS;
```

```

// Lanza la aplicación
public static void main(String[] args) {
    try {
        UIManager.setLookAndFeel(
            "com.sun.java.swing.plaf.windows.WindowsLookAndFeel");
    }
    catch (Throwable e) {
        e.printStackTrace();
    }
    EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                Hijos frame = new Hijos();
                frame.setVisible(true);
            }
            catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}

// Crea la GUI
public Hijos() {
    setTitle("Hijos");
    setBounds(100, 100, 450, 214);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    getContentPane().setLayout(null);

    Dinero = new JLabel("lblDinero");
    Dinero.setBounds(10, 13, 80, 14);
    getContentPane().add(Dinero);

    txtDinero = new JTextField();
    txtDinero.setBounds(90, 10, 90, 20);
    getContentPane().add(txtDinero);
    txtDinero.setColumns(10);

    btnProcesar = new JButton("Procesar");
    btnProcesar.addActionListener(this);
    btnProcesar.setBounds(335, 9, 89, 23);
    getContentPane().add(btnProcesar);

    btnBorrar = new JButton("Borrar");
    btnBorrar.addActionListener(this);
    btnBorrar.setBounds(246, 9, 89, 23);
    getContentPane().add(btnBorrar);

    scpScroll = new JScrollPane();
    scpScroll.setBounds(10, 44, 414, 120);
    getContentPane().add(scpScroll);

    txtS = new JTextArea();
    txtS.setFont(new Font("Monospaced", Font.PLAIN, 12));
    scpScroll.setViewportView(txtS);
}

// Direcciona eventos de tipo ActionEvent
public void actionPerformed(ActionEvent arg0) {

```

```

        if (arg0.getSource() == btnProcesar) {
            actionPerformedBtnProcesar(arg0);
        }
        if (arg0.getSource() == btnBorrar) {
            actionPerformedBtnBorrar(arg0);
        }
    }

    // Procesa la pulsación del botón Procesar
    protected void actionPerformedBtnProcesar(ActionEvent arg0) {
        // Declaración de variables
        double dinero, dinTamar, dinJosue, dinCaleb, dinDaniel, dinDavid;

        // Entrada de datos
        dinero = Double.parseDouble(txtDinero.getText());

        // Proceso de cálculo
        dinJosue = 0.27*dinero;
        dinTamar = 0.85*dinJosue;
        dinDaniel = 0.25*dinero;
        dinCaleb = 0.23*(dinJosue + dinDaniel);
        dinDavid = dinero - (dinTamar + dinJosue + dinCaleb + dinDaniel);

        // Salida de resultados
        txtS.setText("Dinero de Josué : " + dinJosue + "\n");
        txtS.append("Dinero de Tamar : " + dinTamar + "\n");
        txtS.append("Dinero de Daniel : " + dinDaniel + "\n");
        txtS.append("Dinero de Caleb : " + dinCaleb + "\n");
        txtS.append("Dinero de David : " + dinDavid);
    }

    // Procesa la pulsación del botón Borrar
    protected void actionPerformedBtnBorrar(ActionEvent arg0) {
        txtDinero.setText("");
        txtS.setText("");
        txtDinero.requestFocus();
    }
}

```

## Problema 5

Una tienda ha puesto en oferta la venta de un producto ofreciendo un 11% de descuento sobre el importe de la compra. El importe de la compra se calcula multiplicando el precio del producto por la cantidad de unidades adquiridas. El importe a pagar se calcula restando el importe de la compra menos el importe del descuento. Como incentivo especial, la tienda obsequia 2 caramelos por cada unidad adquirida. Dado el precio del producto y la cantidad de unidades adquiridas, diseñe un algoritmo que determine el importe de la compra, el importe del descuento y el importe a pagar.

### Algoritmo

#### Inicio

```

// Declaración de variables
real pre, impcom, impdes, imppag
entero can, car

// Entrada de datos
Leer pre, can

```

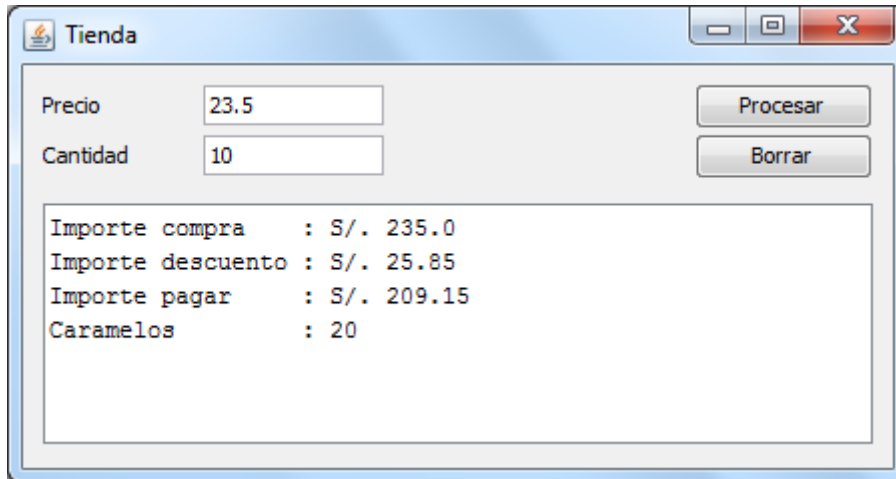
```

// Cálculo de importes
impcom = can*pre
impdes = 0.11*impcom
imppag = impcom - impdes
car = 2*can

// Salida de resultados
Imprimir impcom, impdes, imppag, car
Fin

```

## Programa



```

package cibertec;

import java.awt.EventQueue;
import java.awt.Font;
import javax.swing.JFrame;
import javax.swing.JButton;
import javax.swing.UIManager;
import javax.swing.JLabel;
import javax.swing.JTextField;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;

public class Tienda extends JFrame implements ActionListener {
    private static final long serialVersionUID = 1L;

    // Declaración de variables
    private JLabel lblPrecio;
    private JLabel lblCantidad;
    private JTextField txtPrecio;
    private JTextField txtCantidad;
    private JButton btnProcesar;
    private JButton btnBorrar;
    private JScrollPane scpScroll;
    private JTextArea txtS;

    // Lanza la aplicación
    public static void main(String[] args) {
        try {

```

```

        UIManager.setLookAndFeel(
            "com.sun.java.swing.plaf.windows.WindowsLookAndFeel");
    }
    catch (Throwable e) {
        e.printStackTrace();
    }
    EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                Tienda frame = new Tienda();
                frame.setVisible(true);
            }
            catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}

// Crea la GUI
public Tienda() {
    setTitle("Tienda");
    setBounds(100, 100, 450, 239);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    getContentPane().setLayout(null);

    lblPrecio = new JLabel("Precio");
    lblPrecio.setBounds(10, 13, 80, 14);
    getContentPane().add(lblPrecio);

    lblCantidad = new JLabel("Cantidad");
    lblCantidad.setBounds(10, 38, 80, 14);
    getContentPane().add(lblCantidad);

    txtPrecio = new JTextField();
    txtPrecio.setBounds(90, 10, 90, 20);
    getContentPane().add(txtPrecio);
    txtPrecio.setColumns(10);

    txtCantidad = new JTextField();
    txtCantidad.setBounds(90, 35, 90, 20);
    getContentPane().add(txtCantidad);
    txtCantidad.setColumns(10);

    btnProcesar = new JButton("Procesar");
    btnProcesar.addActionListener(this);
    btnProcesar.setBounds(335, 9, 89, 23);
    getContentPane().add(btnProcesar);

    btnBorrar = new JButton("Borrar");
    btnBorrar.addActionListener(this);
    btnBorrar.setBounds(335, 34, 89, 23);
    getContentPane().add(btnBorrar);

    scpScroll = new JScrollPane();
    scpScroll.setBounds(10, 69, 414, 120);
    getContentPane().add(scpScroll);

    txtS = new JTextArea();

```



```

        txtS.setFont(new Font("Monospaced", Font.PLAIN, 12));
        scpScroll.setViewportView(txtS);
    }

    // Direcciona eventos de tipo ActionEvent
    public void actionPerformed(ActionEvent arg0) {
        if (arg0.getSource() == btnProcesar) {
            actionPerformedBtnProcesar(arg0);
        }
        if (arg0.getSource() == btnBorrar) {
            actionPerformedBtnBorrar(arg0);
        }
    }

    // Procesa la pulsación del botón Procesar
    protected void actionPerformedBtnProcesar(ActionEvent arg0) {
        // Declaración de variables
        int can, car;
        double pre, impcom, impdes, imppag;

        // Entrada de datos
        pre = Double.parseDouble(txtPrecio.getText());
        can = Integer.parseInt(txtCantidad.getText());

        // Proceso de cálculo
        impcom = pre*can;
        impdes = 0.11*impcom;
        imppag = impcom - impdes;
        car = 2*can;

        // Salida de resultados
        txtS.setText("Importe compra      : S/. " + impcom + "\n");
        txtS.append("Importe descuento : S/. " + impdes + "\n");
        txtS.append("Importe pagar      : S/. " + imppag + "\n");
        txtS.append("Caramelos          : " + car);
    }

    // Procesa la pulsación del botón Borrar
    protected void actionPerformedBtnBorrar(ActionEvent arg0) {
        txtPrecio.setText("");
        txtCantidad.setText("");
        txtS.setText("");
        txtPrecio.requestFocus();
    }
}

```

## Problema 6

Una tienda ha puesto en oferta la venta de camisas ofreciendo un descuento, por temporada de verano, denominado 7% + 7%. Los cálculos se efectúan de la siguiente manera:

- El importe de la compra es igual al producto del precio de la camisa por la cantidad de unidades adquiridas.
- El importe del primer descuento es igual al 7% del importe de la compra.
- El importe del segundo descuento es igual al 7% del lo que queda de restar el importe de la compra menos el importe del primer descuento.
- El importe del descuento total es igual a la suma de los dos descuentos anteriores.

- El importe a pagar es igual al importe de la compra menos el importe del descuento total.

Dado el precio del producto y la cantidad de unidades adquiridas, diseñe un programa que determine el importe de la compra, el importe del primer descuento, el importe del segundo descuento, el importe del descuento total y el importe a pagar

### Algoritmo

#### Inicio

```
// Declaración de variables
entero can
real pre, impcom, impdes, imppag, des1, des2

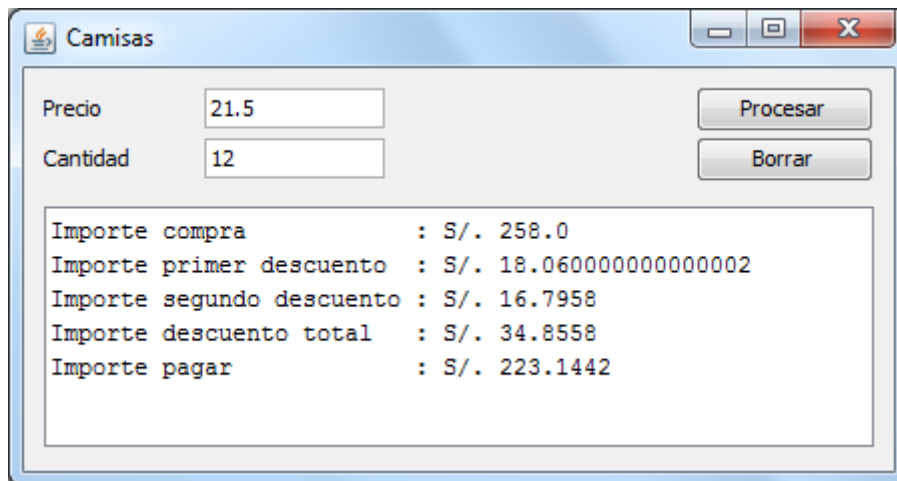
// Entrada de datos
Leer pre, can

// Proceso de cálculo
impcom = pre*can
des1 = 0.07*impcom
des2 = 0.07*(impcom - des1)
impdes = des1 + des2
imppag = impcom - impdes

// Salida de resultados
Imprimir impcom, des1, des2, impdes, imppag
```

#### Fin

### Programa



```
package cibertec;
```

```
import java.awt.EventQueue;
import java.awt.Font;
import javax.swing.JFrame;
import javax.swing.JButton;
import javax.swing.UIManager;
import javax.swing.JLabel;
import javax.swing.JTextField;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.JScrollPane;
```

```

import javax.swing.JTextArea;

public class Camisas extends JFrame implements ActionListener {
    private static final long serialVersionUID = 1L;

    // Declaración de variables
    private JLabel lblPrecio;
    private JLabel lblCantidad;
    private JTextField txtPrecio;
    private JTextField txtCantidad;
    private JButton btnProcesar;
    private JButton btnBorrar;
    private JScrollPane scpScroll;
    private JTextArea txtS;

    // Lanza la aplicación
    public static void main(String[] args) {
        try {
            UIManager.setLookAndFeel(
                "com.sun.java.swing.plaf.windows.WindowsLookAndFeel");
        }
        catch (Throwable e) {
            e.printStackTrace();
        }
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    Camisas frame = new Camisas();
                    frame.setVisible(true);
                }
                catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    // Crea la GUI
    public Camisas() {
        setTitle("Camisas");
        setBounds(100, 100, 450, 239);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        getContentPane().setLayout(null);

        lblPrecio = new JLabel("Precio");
        lblPrecio.setBounds(10, 13, 80, 14);
        getContentPane().add(lblPrecio);

        lblCantidad = new JLabel("Cantidad");
        lblCantidad.setBounds(10, 38, 80, 14);
        getContentPane().add(lblCantidad);

        txtPrecio = new JTextField();
        txtPrecio.setBounds(90, 10, 90, 20);
        getContentPane().add(txtPrecio);
        txtPrecio.setColumns(10);

        txtCantidad = new JTextField();
        txtCantidad.setBounds(90, 35, 90, 20);
    }

```

```

getContentPane().add(txtCantidad);
txtCantidad.setColumns(10);

btnProcesar = new JButton("Procesar");
btnProcesar.addActionListener(this);
btnProcesar.setBounds(335, 9, 89, 23);
getContentPane().add(btnProcesar);

btnBorrar = new JButton("Borrar");
btnBorrar.addActionListener(this);
btnBorrar.setBounds(335, 34, 89, 23);
getContentPane().add(btnBorrar);

scpScroll = new JScrollPane();
scpScroll.setBounds(10, 69, 414, 120);
getContentPane().add(scpScroll);

txtS = new JTextArea();
txtS.setFont(new Font("Monospaced", Font.PLAIN, 12));
scpScroll.setViewportView(txtS);
}

// Direcciona eventos de tipo ActionEvent
public void actionPerformed(ActionEvent arg0) {
    if (arg0.getSource() == btnProcesar) {
        actionPerformedBtnProcesar(arg0);
    }
    if (arg0.getSource() == btnBorrar) {
        actionPerformedBtnBorrar(arg0);
    }
}

// Procesa la pulsación del botón Procesar
protected void actionPerformedBtnProcesar(ActionEvent arg0) {
    // Declaración de variables
    int can;
    double pre, impcom, impdes, imppag, des1, des2;

    // Entrada de datos
    pre = Double.parseDouble(txtPrecio.getText());
    can = Integer.parseInt(txtCantidad.getText());

    // Proceso de cálculo
    impcom = pre*can;
    des1 = 0.07*impcom;
    des2 = 0.07*(impcom - des1);
    impdes = des1 + des2;
    imppag = impcom - impdes;

    // Salida de resultados
    txtS.setText("Importe compra           : S/. " + impcom + "\n");
    txtS.append("Importe primer descuento : S/. " + des1 + "\n");
    txtS.append("Importe segundo descuento : S/. " + des2 + "\n");
    txtS.append("Importe descuento total   : S/. " + impdes + "\n");
    txtS.append("Importe pagar             : S/. " + imppag);
}

// Procesa la pulsación del botón Borrar
protected void actionPerformedBtnBorrar(ActionEvent arg0) {

```

```

        txtPrecio.setText("");
        txtCantidad.setText("");
        txtS.setText("");
        txtPrecio.requestFocus();
    }
}

```

## Problema 7

El cálculo del pago mensual de un empleado de una empresa se efectúa de la siguiente manera:

- Sueldo básico: horas trabajadas por la tarifa horaria
- Bonificación: 20% del sueldo básico
- Sueldo bruto: sueldo básico más bonificación
- Descuento: 10% del sueldo bruto
- Sueldo neto: sueldo bruto menos descuento

Dadas las horas trabajadas y la tarifa horaria de un empleado, diseñe un programa que determine el sueldo básico, la bonificación, el sueldo bruto, el descuento y el sueldo neto que le corresponden.

## Algoritmo

### Inicio

```

// Declaración de variables
real horasTrab, tarifaHor
real sueldoBas, montoBoni, sueldoBru, montoDesc, sueldoNet

// Entrada de datos
Leer horasTrab, tarifaHor

// Proceso de cálculo
sueldoBas = horasTrab*tarifaHor
montoBoni = 0.20*sueldoBas
sueldoBru = sueldoBas+montoBoni
montoDesc = 0.10*sueldoBru
sueldoNet = sueldoBru-montoDesc

// Salida de resultados
Imprimir sueldoBas, montoBoni, sueldoBru, montoDesc, sueldoNet

```

Fin

## Programa

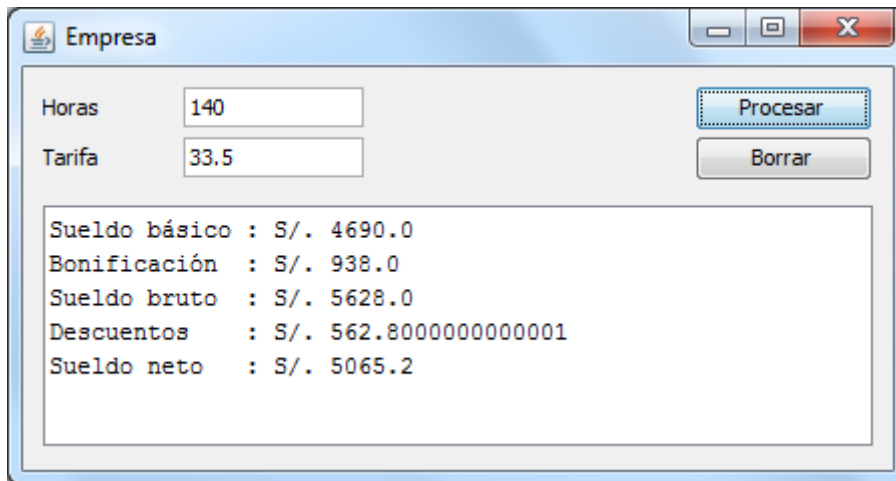
```

package cibertec;

import java.awt.EventQueue;
import java.awt.Font;
import javax.swing.JFrame;
import javax.swing.JButton;
import javax.swing.UIManager;
import javax.swing.JLabel;
import javax.swing.JTextField;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.JScrollPane;

```

```
import javax.swing.JTextArea;
```



```
public class Empresa extends JFrame implements ActionListener {
    private static final long serialVersionUID = 1L;

    // Declaración de variables
    private JLabel lblHoras;
    private JLabel lblTarifa;
    private JTextField txtHoras;
    private JTextField txtTarifa;
    private JButton btnProcesar;
    private JButton btnBorrar;
    private JScrollPane scpScroll;
    private JTextArea txtS;

    // Lanza la aplicación
    public static void main(String[] args) {
        try {
            UIManager.setLookAndFeel(
                "com.sun.java.swing.plaf.windows.WindowsLookAndFeel");
        }
        catch (Throwable e) {
            e.printStackTrace();
        }
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    Empresa frame = new Empresa();
                    frame.setVisible(true);
                }
                catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    // Crea la GUI
    public Empresa() {
        setTitle("Empresa");
        setBounds(100, 100, 450, 239);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        getContentPane().setLayout(null);
    }
}
```

```

lblHoras = new JLabel("Horas");
lblHoras.setBounds(10, 13, 70, 14);
getContentPane().add(lblHoras);

lblTarifa = new JLabel("Tarifa");
lblTarifa.setBounds(10, 38, 70, 14);
getContentPane().add(lblTarifa);

txtHoras = new JTextField();
txtHoras.setBounds(80, 10, 90, 20);
getContentPane().add(txtHoras);
txtHoras.setColumns(10);

txtTarifa = new JTextField();
txtTarifa.setBounds(80, 35, 90, 20);
getContentPane().add(txtTarifa);
txtTarifa.setColumns(10);

btnProcesar = new JButton("Procesar");
btnProcesar.addActionListener(this);
btnProcesar.setBounds(335, 9, 89, 23);
getContentPane().add(btnProcesar);

btnBorrar = new JButton("Borrar");
btnBorrar.addActionListener(this);
btnBorrar.setBounds(335, 34, 89, 23);
getContentPane().add(btnBorrar);

scpScroll = new JScrollPane();
scpScroll.setBounds(10, 69, 414, 120);
getContentPane().add(scpScroll);

txtS = new JTextArea();
txtS.setFont(new Font("Monospaced", Font.PLAIN, 12));
scpScroll.setViewportView(txtS);
}

// Direcciona eventos de tipo ActionEvent
public void actionPerformed(ActionEvent arg0) {
    if (arg0.getSource() == btnProcesar) {
        actionPerformedBtnProcesar(arg0);
    }
    if (arg0.getSource() == btnBorrar) {
        actionPerformedBtnBorrar(arg0);
    }
}

// Procesa la pulsación del botón Procesar
protected void actionPerformedBtnProcesar(ActionEvent arg0) {
    // Declaración de variables
    double horasTrab, tarifaHor;
    double sueldoBas, montoBoni, sueldoBru, montoDesc, sueldoNet;

    // Entrada de datos
    horasTrab = Double.parseDouble(txtHoras.getText());
    tarifaHor = Double.parseDouble(txtTarifa.getText());

    // Cálculo de montos

```

```

        sueldoBas = horasTrab*tarifaHor;
        montoBoni = 0.20*sueldoBas;
        sueldoBru = sueldoBas+montoBoni;
        montoDesc = 0.10*sueldoBru;
        sueldoNet = sueldoBru-montoDesc;

        // Salida de resultados
        txtS.setText("Sueldo básico : S/. " + sueldoBas + "\n");
        txtS.append("Bonificación : S/. " + montoBoni + "\n");
        txtS.append("Sueldo bruto : S/. " + sueldoBru + "\n");
        txtS.append("Descuentos : S/. " + montoDesc + "\n");
        txtS.append("Sueldo neto : S/. " + sueldoNet);
    }

    // Procesa la pulsación del botón Borrar
    protected void actionPerformedBtnBorrar(ActionEvent arg0) {
        txtHoras.setText("");
        txtTarifa.setText("");
        txtS.setText("");
        txtHoras.requestFocus();
    }
}

```

## Problema 8

Dadas las cantidades de dinero aportadas por Débora, Raquel y Séfora para formar un capital, diseñe un programa que determine el monto del capital formado y el porcentaje de dicho capital que aporta cada una.

### Algoritmo

#### Inicio

```

// Declaración de variables
real dineDeb, dineRaq, dineSef, capital, porcDeb, porcRaq, porcSef

// Entrada de datos
Leer dineDeb, dineRaq, dineSef

// Determina el capital formado
capital = dineDeb + dineRaq + dineSef

// Determina los porcentajes
porcDeb = dineDeb*100/capital
porcRaq = dineRaq*100/capital
porcSef = dineSef*100/capital

// Salida de resultados
Imprimir capital, porcDeb, porcRaq, porcSef

```

Fin

### Programa

```

package cibertec;

import java.awt.EventQueue;
import java.awt.Font;
import javax.swing.JFrame;
import javax.swing.JButton;

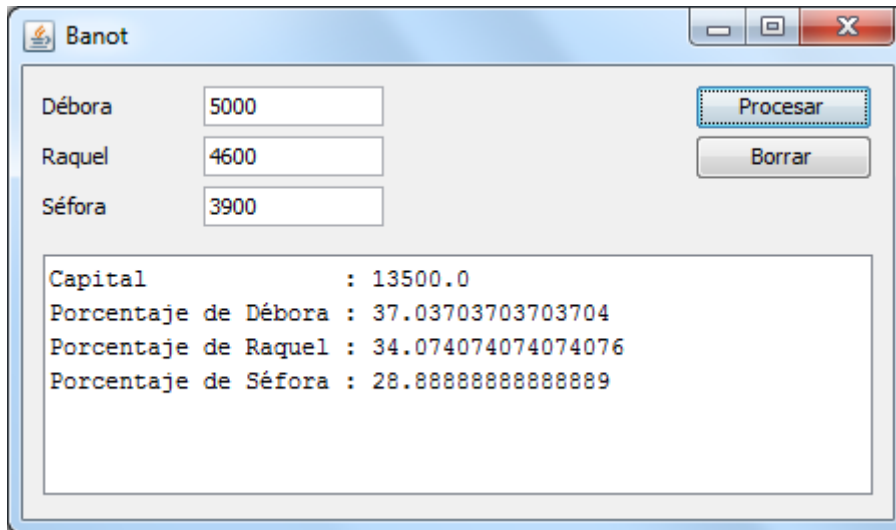
```



```

import javax.swing.UIManager;
import javax.swing.JLabel;
import javax.swing.JTextField;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;

```



```

public class Banot extends JFrame implements ActionListener {
    private static final long serialVersionUID = 1L;

    // Declaración de variables
    private JLabel lblDebora;
    private JLabel lblRaquel;
    private JLabel lblSefora;
    private JTextField txtDebora;
    private JTextField txtRaquel;
    private JTextField txtSefora;
    private JButton btnProcesar;
    private JButton btnBorrar;
    private JScrollPane scpScroll;
    private JTextArea txtS;

    // Lanza la aplicación
    public static void main(String[] args) {
        try {
            UIManager.setLookAndFeel(
                "com.sun.java.swing.plaf.windows.WindowsLookAndFeel");
        }
        catch (Throwable e) {
            e.printStackTrace();
        }
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    Banot frame = new Banot();
                    frame.setVisible(true);
                }
                catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }
}

```

```

    }
    });
}

// Crea la GUI
public Banot() {
    setTitle("Banot");
    setBounds(100, 100, 450, 264);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    getContentPane().setLayout(null);

    lblDebora = new JLabel("Débora");
    lblDebora.setBounds(10, 13, 80, 14);
    getContentPane().add(lblDebora);

    lblRaquel = new JLabel("Raquel");
    lblRaquel.setBounds(10, 38, 80, 14);
    getContentPane().add(lblRaquel);

    lblSefora = new JLabel("Séfora");
    lblSefora.setBounds(10, 63, 80, 14);
    getContentPane().add(lblSefora);

    txtDebora = new JTextField();
    txtDebora.setBounds(90, 10, 90, 20);
    getContentPane().add(txtDebora);
    txtDebora.setColumns(10);

    txtRaquel = new JTextField();
    txtRaquel.setBounds(90, 35, 90, 20);
    getContentPane().add(txtRaquel);
    txtRaquel.setColumns(10);

    txtSefora = new JTextField();
    txtSefora.setBounds(90, 60, 90, 20);
    getContentPane().add(txtSefora);
    txtSefora.setColumns(10);

    btnProcesar = new JButton("Procesar");
    btnProcesar.addActionListener(this);
    btnProcesar.setBounds(335, 9, 89, 23);
    getContentPane().add(btnProcesar);

    btnBorrar = new JButton("Borrar");
    btnBorrar.addActionListener(this);
    btnBorrar.setBounds(335, 34, 89, 23);
    getContentPane().add(btnBorrar);

    scpScroll = new JScrollPane();
    scpScroll.setBounds(10, 94, 414, 120);
    getContentPane().add(scpScroll);

    txtS = new JTextArea();
    txtS.setFont(new Font("Monospaced", Font.PLAIN, 12));
    scpScroll.setViewportView(txtS);
}

// Direcciona eventos de tipo ActionEvent
public void actionPerformed(ActionEvent arg0) {

```

```

        if (arg0.getSource() == btnProcesar) {
            actionPerformedBtnProcesar(arg0);
        }
        if (arg0.getSource() == btnBorrar) {
            actionPerformedBtnBorrar(arg0);
        }
    }

    // Procesa la pulsación del botón Procesar
    protected void actionPerformedBtnProcesar(ActionEvent arg0) {
        // Declaración de variables
        double dineDeb, dineRaq, dineSef, capital, porcDeb, porcRaq, porcSef;

        // Entrada de datos
        dineDeb = Double.parseDouble(txtDebora.getText());
        dineRaq = Double.parseDouble(txtRaquel.getText());
        dineSef = Double.parseDouble(txtSefora.getText());

        // Determina el capital formado
        capital = dineDeb + dineRaq + dineSef;

        // Determina los porcentajes
        porcDeb = dineDeb*100/capital;
        porcRaq = dineRaq*100/capital;
        porcSef = dineSef*100/capital;

        // Salida de resultados
        txtS.setText("Capital          : " + capital + "\n");
        txtS.append("Porcentaje de Débora : " + porcDeb + "\n");
        txtS.append("Porcentaje de Raquel : " + porcRaq + "\n");
        txtS.append("Porcentaje de Séfora : " + porcSef);
    }

    // Procesa la pulsación del botón Borrar
    protected void actionPerformedBtnBorrar(ActionEvent arg0) {
        txtDebora.setText("");
        txtRaquel.setText("");
        txtSefora.setText("");
        txtS.setText("");
        txtDebora.requestFocus();
    }
}

```

## Problema 9

Dada una cantidad de dinero y las edades de tres personas, diseñe un programa que reparta el dinero en forma proporcional a las edades. El monto que le corresponde a cada persona se calcula con la siguiente fórmula:

$$\text{monto de la persona} = \frac{\text{edad de la persona} \times \text{dinero a repartir}}{\text{suma total de edades}}$$

## Algoritmo

### Inicio

```

// Declaración de variables
real montoP1, montoP2, montoP3, montoRepartir
entero edadP1, edadP2, edadP3, sumaEdades

```

```

// Entrada de datos
Leer montoRepartir, edadP1, edadP2, edadP3

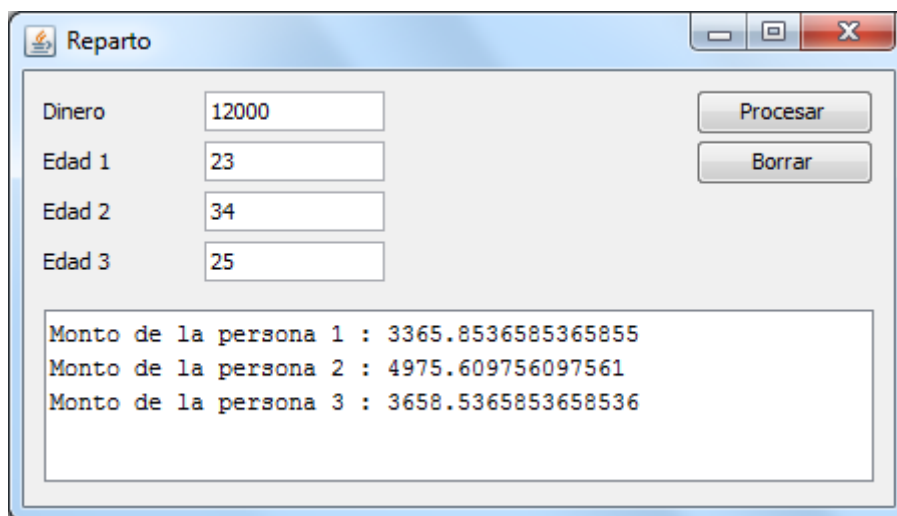
// Calcula la suma total de edades
sumaEdades = edadP1 + edadP2 + edadP3

// Calcula la cantidad de dinero de cada persona
montoP1 = (edadP1 * montoRepartir) / sumaEdades
montoP2 = (edadP2 * montoRepartir) / sumaEdades
montoP3 = (edadP3 * montoRepartir) / sumaEdades

// Salida de resultados
Imprimir montoP1, montoP2, montoP3
Fin

```

## Programa



```

package cibertec;

import java.awt.EventQueue;
import java.awt.Font;
import javax.swing.JFrame;
import javax.swing.JButton;
import javax.swing.UIManager;
import javax.swing.JLabel;
import javax.swing.JTextField;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;

public class Reparto extends JFrame implements ActionListener {
    private static final long serialVersionUID = 1L;

    // Declaración de variables
    private JLabel lblDinero;
    private JLabel lblEdad1;
    private JLabel lblEdad2;
    private JTextField txtDinero;
    private JTextField txtEdad1;
    private JTextField txtEdad2;

```

```

private JButton btnProcesar;
private JButton btnBorrar;
private JScrollPane scpScroll;
private JTextArea txtS;
private JLabel lblEdad3;
private JTextField txtEdad3;

// Lanza la aplicación
public static void main(String[] args) {
    try {
        UIManager.setLookAndFeel(
            "com.sun.java.swing.plaf.windows.WindowsLookAndFeel");
    }
    catch (Throwable e) {
        e.printStackTrace();
    }
    EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                Reparto frame = new Reparto();
                frame.setVisible(true);
            }
            catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}

// Crea la GUI
public Reparto() {
    setTitle("Reparto");
    setBounds(100, 100, 450, 255);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    getContentPane().setLayout(null);

    lblDinero = new JLabel("Dinero");
    lblDinero.setBounds(10, 13, 80, 14);
    getContentPane().add(lblDinero);

    lblEdad1 = new JLabel("Edad 1");
    lblEdad1.setBounds(10, 38, 80, 14);
    getContentPane().add(lblEdad1);

    lblEdad2 = new JLabel("Edad 2");
    lblEdad2.setBounds(10, 63, 80, 14);
    getContentPane().add(lblEdad2);

    lblEdad3 = new JLabel("Edad 3");
    lblEdad3.setBounds(10, 88, 80, 14);
    getContentPane().add(lblEdad3);

    txtDinero = new JTextField();
    txtDinero.setBounds(90, 10, 90, 20);
    getContentPane().add(txtDinero);
    txtDinero.setColumns(10);

    txtEdad1 = new JTextField();
    txtEdad1.setBounds(90, 35, 90, 20);

```

```

getContentPane().add(txtEdad1);
txtEdad1.setColumns(10);

txtEdad2 = new JTextField();
txtEdad2.setBounds(90, 60, 90, 20);
getContentPane().add(txtEdad2);
txtEdad2.setColumns(10);

txtEdad3 = new JTextField();
txtEdad3.setColumns(10);
txtEdad3.setBounds(90, 85, 90, 20);
getContentPane().add(txtEdad3);

btnProcesar = new JButton("Procesar");
btnProcesar.addActionListener(this);
btnProcesar.setBounds(335, 9, 89, 23);
getContentPane().add(btnProcesar);

btnBorrar = new JButton("Borrar");
btnBorrar.addActionListener(this);
btnBorrar.setBounds(335, 34, 89, 23);
getContentPane().add(btnBorrar);

scpScroll = new JScrollPane();
scpScroll.setBounds(10, 119, 414, 86);
getContentPane().add(scpScroll);

txtS = new JTextArea();
txtS.setFont(new Font("Monospaced", Font.PLAIN, 12));
scpScroll.setViewportView(txtS);
}

// Direcciona eventos de tipo ActionEvent
public void actionPerformed(ActionEvent arg0) {
    if (arg0.getSource() == btnProcesar) {
        actionPerformedBtnProcesar(arg0);
    }
    if (arg0.getSource() == btnBorrar) {
        actionPerformedBtnBorrar(arg0);
    }
}

// Procesa la pulsación del botón Procesar
protected void actionPerformedBtnProcesar(ActionEvent arg0) {
    // Declaración de variables
    double monto1, monto2, monto3, dinero;
    int edad1, edad2, edad3, sumaEdades;

    // Entrada de datos
    dinero = Double.parseDouble(txtDinero.getText());
    edad1 = Integer.parseInt(txtEdad1.getText());
    edad2 = Integer.parseInt(txtEdad2.getText());
    edad3 = Integer.parseInt(txtEdad3.getText());

    // Calcula la suma total de edades
    sumaEdades = edad1 + edad2 + edad3;

    // Calcula la cantidad de dinero de cada persona
    monto1 = (edad1 * dinero) / sumaEdades;

```

```
    monto2 = (edad2 * dinero) / sumaEdades;
    monto3 = (edad3 * dinero) / sumaEdades;

    // Salida de resultados
    txtS.setText("Monto de la persona 1 : " + monto1 + "\n");
    txtS.append("Monto de la persona 2 : " + monto2 + "\n");
    txtS.append("Monto de la persona 3 : " + monto3);
}

// Procesa la pulsación del botón Borrar
protected void actionPerformedBtnBorrar(ActionEvent arg0) {
    txtDinero.setText("");
    txtEdad1.setText("");
    txtEdad2.setText("");
    txtEdad3.setText("");
    txtS.setText("");
    txtDinero.requestFocus();
}
}
```

# Problemas propuestos

## Actividades

1. Diseñe un algoritmo que determine el área lateral ( $AL$ ), el área total ( $AT$ ) y el área de la base ( $AB$ ) de un cilindro del que se conoce su radio ( $r$ ) y su altura ( $h$ ). Considere las siguientes fórmulas:

$$AT = 2 \times AB + AL$$

$$AB = 3.1416 \times r^2$$

$$AL = 2 \times 3.1416 \times r \times h$$

2. Diseñe un algoritmo que determine el área de la base ( $AB$ ), el perímetro de la base ( $PB$ ) y el área lateral ( $AL$ ) de un cono del que se conoce el radio de su base ( $R$ ) y su generatriz ( $G$ ). Considere las siguientes fórmulas:

$$AB = 3.14 \times R^2$$

$$AL = \frac{PB \times G}{2}$$

$$PB = 6.28 \times R$$

3. Diseñe un algoritmo que determine el área ( $A$ ) y el volumen ( $V$ ) de una esfera de la que se conoce su radio ( $r$ ). Considere las siguientes fórmulas:

$$A = 12.57 \times r^2$$

$$V = \frac{12.57 \times r^3}{3}$$

4. Una institución social ha recibido una donación en dinero que lo repartirá entre cinco áreas. Cada área recibirá una parte de la donación equivalente a:

- Centro de salud: 25% de la donación
- Comedor: 45% del monto recibido por la escuela
- Biblioteca: 17% del monto total recibido entre el comedor y la escuela
- Escuela: 35% de la donación
- Biblioteca: 40% del monto recibido por el centro de salud
- Asilo de ancianos: lo que queda de la donación

Dado el importe de la donación, diseñe un algoritmo que determine qué cantidad de dinero le corresponde a cada área.

5. Una empresa ha recibido una donación en dinero que lo repartirá entre cinco áreas. Cada área recibirá una parte de la donación equivalente a:

- Área de producción: 25% del monto de la donación
- Área de contabilidad: 40% del monto total recibido entre las áreas de marketing y soporte



- Área de marketing: 15% del monto total recibido entre las áreas de producción y soporte
- Área de soporte: 20% del monto de la donación
- Área de recursos humanos: lo que queda del monto de la donación

Dado el importe de la donación, diseñe un algoritmo que determine el monto de dinero que recibirá cada área.

6. Una tienda ha puesto en oferta la venta de un producto ofreciendo un descuento igual al 15% del importe de la compra. El importe de la compra se calcula multiplicando el precio del producto por la cantidad de unidades adquiridas. El importe a pagar se calcula restando el importe de la compra menos el importe del descuento. Dado el precio del producto y la cantidad de unidades adquiridas, diseñe un algoritmo que determine el importe de la compra, el importe del descuento y el importe a pagar.
7. Una empresa de transportes aplica un descuento igual al 7% del importe de la compra. El importe de la compra se calcula multiplicando el precio del pasaje por la cantidad de pasajes adquiridos. El importe a pagar se calcula restando el importe de la compra menos el importe del descuento. Como incentivo adicional, la empresa obsequia 3 chocolates por cada pasaje adquirido. Dado el precio del pasaje y la cantidad de pasajes adquiridos, diseñe un algoritmo que determine el importe de la compra, el importe del descuento, el importe a pagar y la cantidad de chocolates de obsequio que le corresponden a un cliente.
8. Una imprenta ha lanzado al mercado la venta al por mayor del cuaderno de 100 hojas *mentor* que es distribuido a nivel nacional. El importe compra se calcula multiplicando el precio de la docena por la cantidad de docenas adquiridas. Como oferta, la imprenta aplica un descuento del 12% del importe compra. El importe a pagar se calcula restando el importe de la compra menos el importe del descuento y sumando el costo del transporte. Dado el precio de la docena, la cantidad de docenas adquiridas y el costo del transporte, diseñe un algoritmo que determine el importe compra, el importe del descuento y el importe a pagar que le corresponden a un cliente.
9. Una empresa paga a sus empleados por horas trabajadas. El sueldo bruto se calcula multiplicando las horas trabajadas por la tarifa horaria del empleado. Por ley, todo empleado esta sujeto a un descuento igual al 15% del sueldo bruto. El sueldo neto se calcula restando el sueldo bruto menos el importe del descuento. Dado el número de horas trabajadas y la tarifa horaria de un empleado, diseñe un algoritmo que determine el sueldo bruto, el descuento y el sueldo neto del empleado.
10. Una empresa paga a sus vendedores un sueldo bruto que se calcula sumando un sueldo básico más una comisión. El sueldo básico es S/. 350.75. La comisión es igual al 5% del importe total vendido en el mes. Por ley, todo empleado está sujeto a un descuento igual al 15% del sueldo bruto. El sueldo neto se calcula restando el sueldo bruto menos el importe del descuento. Dado el importe total vendido en el mes, diseñe un algoritmo que imprima la boleta de un vendedor indicando el sueldo básico, la comisión, el sueldo bruto, el descuento y el sueldo neto.

## Autoevaluación

1. Diseñe un algoritmo que lea un ángulo en grados sexagesimales (S) y lo convierta a sus equivalentes en grados centesimales (C) y radianes (R). Considere las siguientes fórmulas:

$$C = \frac{200 \times S}{180}$$

$$R = \frac{3.1416 \times S}{180}$$

2. Diseñe un algoritmo que lea una temperatura en grados Centígrados (C) y la convierta a sus equivalentes en grados Fahrenheit (F), grados Kelvin (K) y grados Rankine(R). Utilice las siguientes fórmulas:

$$K = R - 187$$

$$R = C + 460$$

$$F = \frac{9C}{5} + 32$$

3. Para estimar el peso de un niño en situaciones de emergencias pediátricas, se utiliza la siguiente fórmula:

$$\text{peso en kilogramos} = 3 \times \text{edad en años} + 7$$

Dada la edad de un niño en años, diseñe un algoritmo que determine el peso estimado del niño.

4. Un hospital ha recibido una donación especial que será repartida entre las áreas de Pediatría, Medicina General, Ginecología y Traumatología. Cada área recibirá una parte de la donación equivalente a:

- Pediatría: 20% del monto total recibido entre Medicina General y Ginecología
- Medicina General: 45% de la donación
- Ginecología: 80% del monto recibido por Medicina General
- Traumatología: lo que resta la donación

Dado el monto de la donación, diseñe un algoritmo que determine cuánto recibirá cada área

5. Un padre desea repartir una cantidad de dinero entre sus tres hijos. Cada uno recibirá una parte del dinero equivalente a:

- Juan: 45% del dinero a repartir
- Pedro: 60% del monto recibido por Juan
- Luis: Lo que queda del monto de dinero a repartir

Dado el monto de dinero a repartir, diseñe un algoritmo que determine que cantidad de dinero le corresponde a cada hijo.

6. Una empresa confecciona polos de tamaño estándar aplicando un descuento del 11.5% del importe de la compra. El importe de la compra se calcula multiplicando el precio del polo por la cantidad de polos adquiridos. El importe a pagar se calcula restando el importe compra menos el importe del descuento. Adicionalmente, la empresa obsequia dos lapiceros por cada polo adquirido. Dado el precio del polo y la cantidad de polos adquiridos, diseñe un algoritmo que determine el importe de la compra, el importe del descuento, el importe a pagar y la cantidad de lapiceros de obsequio que le corresponden a un cliente.
7. Una tienda ha puesto en oferta la venta un producto ofreciendo un descuento denominado 10%+10% que consiste en aplicar dos descuentos del 10%. El importe compra se calcula multiplicando el precio del producto por la cantidad de unidades adquiridas. El primer descuento es igual al 10% del importe compra. El segundo descuento es igual al 10% del importe que queda de restar el importe compra menos el importe del primer descuento. El importe del descuento total se calcula sumando el primer y el segundo descuento. El importe a pagar se calcula restando el importe compra menos el importe del descuento total. Dado el precio del producto y la cantidad de unidades adquiridas, diseñe un algoritmo que determine el importe de la compra, el importe del descuento total y el importe a pagar.
8. Los cálculos salariales de los vendedores de una empresa se efectúan de la siguiente manera:
- Sueldo básico mensual: S/.300
  - Comisión por ventas: 9% del importe total vendido en el mes
  - Sueldo bruto: sueldo básico más comisión
  - Descuento: 11% de sueldo bruto
  - Sueldo neto: sueldo bruto menos descuento

Dado el importe total vendido en el mes, diseñe un algoritmo que imprima un reporte indicando el sueldo básico, la comisión, el sueldo bruto, el descuento y el sueldo neto.

9. Los cálculos salariales de los empleados de una empresa se efectúan de la siguiente manera:
- Sueldo bruto: horas trabajadas por la tarifa horaria del empleado
  - Descuento por ESSALUD: 9% del sueldo bruto
  - Descuento por AFP: 10% del sueldo bruto
  - Descuento total: suma de los descuentos anteriores
  - Sueldo neto: sueldo bruto menos descuento total

Dado el número de horas trabajadas y la tarifa horaria, diseñe un algoritmo que determine el sueldo bruto, el descuento por ESSALUD, el descuento por AFP, el descuento total y el sueldo neto.

10. Una empresa ha decidido otorgar una bonificación a sus empleados por única vez. La bonificación estará compuesta de la suma de una bonificación por hijos más una bonificación por tiempo de servicio. La bonificación por hijos será igual a S/. 25 por cada hijo. La bonificación por tiempo de servicio será igual a S/. 50 por cada año de tiempo de servicio. Dado el número de hijos y el número de años de tiempo de servicio, diseñe un algoritmo que determine el importe de la bonificación por hijos, el importe de la bonificación por tiempo de servicio y el importe de la bonificación total que le corresponden a un empleado.

## Para recordar

- Las instrucciones secuenciales se efectúan de arriba hacia abajo, por lo que si en un cierto punto del programa se requiere el valor de una variable, esta debe haber sido asignada previamente.

# Estructura de selección simple if

---

## LOGRO DE LA UNIDAD DE APRENDIZAJE

Al finalizar la unidad, el alumno, mediante el uso de estructuras de selección, diseña algoritmos que involucren procesos selectivos.

## TEMARIO

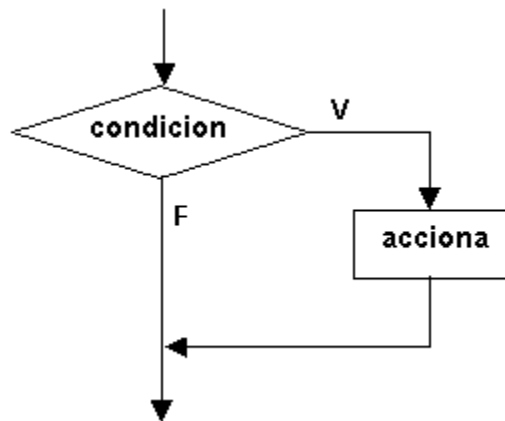
1. Estructura de selección simple if

## ACTIVIDADES

Los alumnos desarrollan algoritmos que involucran la estructuras de selección if.

## 1. Estructura de selección simple if

La *estructura de selección simple if* evalúa una *condición lógica* y, en caso resulte verdadera, efectúa la acción **acciona**. En caso que la condición resulte falsa, continúa con la siguiente instrucción del programa. La acción **acciona** puede ser una **acción simple** (una sola acción) o una **acción compuesta** (bloque de acciones).



**Figura 1** Diagrama de flujo de la estructura de selección simple if

En la tabla que sigue, se muestra el código y el pseudocódigo de la estructura de selección simple if. Note que, en el caso de bloques de acciones (más de una acción), estas deben estar encerradas entre llaves de bloque { }.

| Código Java   | Pseudocódigo   |
|---|--|
| <code>if( condicion )<br/>    accionA;</code>   | <code>si( condicion )<br/>    accionA</code>   |
| <code>if( condicion ){<br/>    accionA1;<br/>    accionA2;<br/>    .<br/>    .<br/>    .<br/>    accionAn;<br/>}</code> | <code>si( condicion ){<br/>    accionA1<br/>    accionA2<br/>    .<br/>    .<br/>    .<br/>    accionAn<br/>}</code> |

## 2. Problemas propuestos

### Problema 1

Una tienda vende un producto a precios unitarios que dependen de la cantidad de unidades adquiridas de acuerdo con la siguiente tabla:

| Unidades adquiridas | Precio unitario |
|---------------------|-----------------|
| 1 a 25              | S/. 27.7        |
| 26 a 50             | S/. 25.5        |
| 51 a 75             | S/. 23.5        |
| 76 en adelante      | S/. 21.5        |

Adicionalmente, si el cliente adquiere más de 50 unidades la tienda le descuenta el 15% del importe de la compra; en caso contrario, sólo le descuenta el 5% del importe compra.

Diseñe un programa que determine el importe de la compra, el importe del descuento y el importe a pagar por la compra de cierta cantidad de unidades del producto.

### Algoritmo

#### Inicio

```
// Declaración de variables
entero unidades;
real impcom = 0, impdes, imppag

// Entrada de datos
Leer unidades

// Calcula el importe de la compra
si(unidades >= 1 && unidades <= 25)
    impcom = unidades*27.5
si(unidades >= 26 && unidades <= 50)
    impcom = unidades*25.5
si(unidades >= 51 && unidades <= 75)
    impcom = unidades*27.5
si(unidades >= 76)
    impcom = unidades*27.5

// Calcula el importe del descuento
si(unidades > 50)
    impdes = 0.15*impcom
sino
    impdes = 0.05*impcom

// Calcula el importe a pagar
imppag = impcom - impdes;

// Salida de resultados
Imprimir impcom, impdes, imppag
```

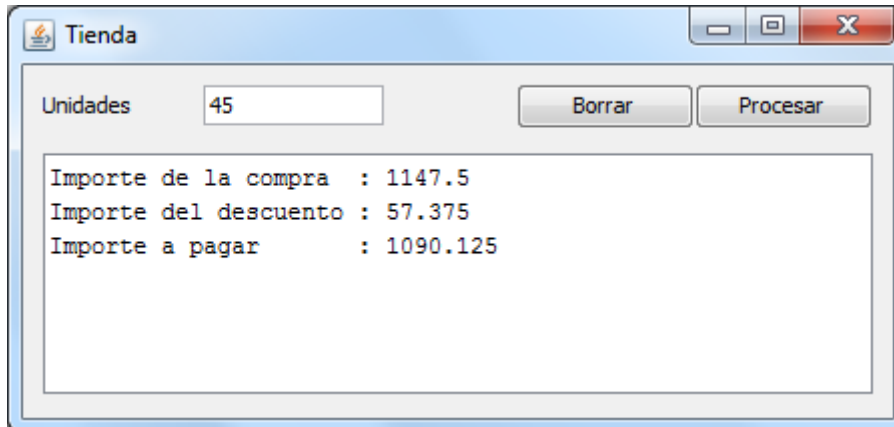
Fin

### Programa

```
package cibertec;

import java.awt.EventQueue;
import javax.swing.JFrame;
import javax.swing.JButton;
import javax.swing.UIManager;
import javax.swing.JLabel;
import javax.swing.JTextField;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
```

```
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
import java.awt.Font;
```



```
public class Tienda extends JFrame implements ActionListener {
    private static final long serialVersionUID = 1L;

    // Declaración de variables
    private JLabel lblUnidades;
    private JTextField txtUnidades;
    private JButton btnProcesar;
    private JButton btnBorrar;
    private JScrollPane scpScroll;
    private JTextArea txtS;

    // Lanza la aplicación
    public static void main(String[] args) {

        try {
            UIManager.setLookAndFeel
                ("com.sun.java.swing.plaf.windows.WindowsLookAndFeel");
        }
        catch (Throwable e) {
            e.printStackTrace();
        }
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    Tienda frame = new Tienda();
                    frame.setVisible(true);
                }
                catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    // Crea la GUI
    public Tienda() {
        setTitle("Tienda");
        setBounds(100, 100, 450, 214);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        getContentPane().setLayout(null);
    }
}
```



```

lblUnidades = new JLabel("Unidades");
lblUnidades.setBounds(10, 13, 80, 14);
getContentPane().add(lblUnidades);

txtUnidades = new JTextField();
txtUnidades.setBounds(90, 10, 90, 20);
getContentPane().add(txtUnidades);
txtUnidades.setColumns(10);

btnProcesar = new JButton("Procesar");
btnProcesar.addActionListener(this);
btnProcesar.setBounds(335, 9, 89, 23);
getContentPane().add(btnProcesar);

btnBorrar = new JButton("Borrar");
btnBorrar.addActionListener(this);
btnBorrar.setBounds(246, 9, 89, 23);
getContentPane().add(btnBorrar);

scpScroll = new JScrollPane();
scpScroll.setBounds(10, 44, 414, 120);
getContentPane().add(scpScroll);

txtS = new JTextArea();
txtS.setFont(new Font("Monospaced", Font.PLAIN, 12));
scpScroll.setViewportView(txtS);
}

// Direcciona eventos de tipo ActionEvent
public void actionPerformed(ActionEvent arg0) {
    if (arg0.getSource() == btnProcesar) {
        actionPerformedBtnProcesar(arg0);
    }
    if (arg0.getSource() == btnBorrar) {
        actionPerformedBtnBorrar(arg0);
    }
}

// Procesa la pulsación del botón Procesar
protected void actionPerformedBtnProcesar(ActionEvent arg0) {
    // Declaración de variables
    int unidades;
    double impcom = 0, impdes, imppag;

    // Entrada de datos
    unidades = Integer.parseInt(txtUnidades.getText());

    // Calcula el importe de la compra
    if (unidades >= 1 && unidades <= 25)
        impcom = unidades * 27.5;
    if (unidades >= 26 && unidades <= 50)
        impcom = unidades * 25.5;
    if (unidades >= 51 && unidades <= 75)
        impcom = unidades * 27.5;
    if (unidades >= 76)
        impcom = unidades * 27.5;

    // Calcula el importe del descuento
    if (unidades > 50)

```

```

        impdes = 0.15 * impcom;
    else
        impdes = 0.05 * impcom;

    // Calcula el importe a pagar
    imppag = impcom - impdes;

    // Salida de resultados
    txtS.setText("Importe de la compra : " + impcom + "\n");
    txtS.append("Importe del descuento : " + impdes + "\n");
    txtS.append("Importe a pagar : " + imppag);

}

// Procesa la pulsación del botón Borrar
protected void actionPerformedBtnBorrar(ActionEvent arg0) {
    txtUnidades.setText("");
    txtS.setText("");
    txtUnidades.requestFocus();
}
}

```

## Problema 2

Una tienda ha decidido incentivar a sus clientes con un obsequio. Para ello, el cliente debe extraer un bolo de una urna que contiene 100 bolos numerados del 1 al 100. Luego, sobre la base del número del bolo se obtiene el obsequio de acuerdo con la siguiente tabla:

| Número del bolo | Obsequio                 |
|-----------------|--------------------------|
| 1 a 20          | Un lapicero              |
| 21 a 40         | Un cuaderno de 100 hojas |
| 41 a 60         | Una caja de plumones     |
| 61 a 80         | Un cuaderno espiral      |
| 81 a 99         | Una agenda               |
| 100             | Una mochila              |

Dado el número del bolo obtenido por un cliente, diseñe un programa que determine qué obsequio le corresponde. En caso que el número ingresado sea incorrecto, como obsequio mostrar "Ninguno".

## Algoritmo

### Inicio

```

// Declaración de variables
entero numero
cadena obsequio = "Ninguno"

// Entrada de datos
Leer numero

// Determina el obsequio
si(numero >= 1 && numero <= 20)
    obsequio = "Un lapicero"
si(numero >= 21 && numero <= 40)
    obsequio = "Un cuaderno de 100 hojas"

```

```

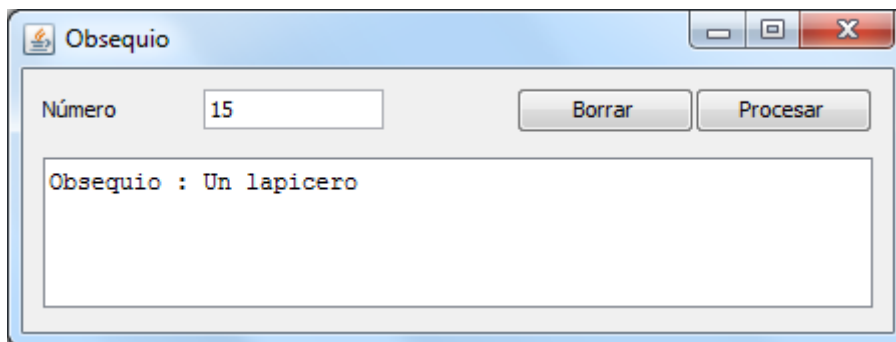
si(numero >= 41 && numero <= 60)
    obsequio = "Una caja de 12 plumones"
si(numero >= 61 && numero <= 80)
    obsequio = "Un cuaderno espiral"
si(numero >= 81 && numero <= 99)
    obsequio = "Una agenda"
si(numero == 100)
    obsequio = "Una mochila"

// Salida de resultados
Imprimir obsequio

```

Fin

## Programa



```

package cibertec;

import java.awt.EventQueue;
import java.awt.Font;
import javax.swing.JFrame;
import javax.swing.JButton;
import javax.swing.UIManager;
import javax.swing.JLabel;
import javax.swing.JTextField;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;

public class Obsequio extends JFrame implements ActionListener {
    private static final long serialVersionUID = 1L;

    // Declaración de variables
    private JLabel lblNumero;
    private JTextField txtNumero;
    private JButton btnProcesar;
    private JButton btnBorrar;
    private JScrollPane scpScroll;
    private JTextArea txtS;

    // Lanza la aplicación
    public static void main(String[] args) {
        try {
            UIManager.setLookAndFeel
                ("com.sun.java.swing.plaf.windows.WindowsLookAndFeel");
        }
    }

```

```

        catch (Throwable e) {
            e.printStackTrace();
        }
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    Obsequio frame = new Obsequio();
                    frame.setVisible(true);
                }
                catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    // Crea la GUI
    public Obsequio() {
        setTitle("Obsequio");
        setBounds(100, 100, 450, 169);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        getContentPane().setLayout(null);

        lblNumero = new JLabel("Número");
        lblNumero.setBounds(10, 13, 80, 14);
        getContentPane().add(lblNumero);

        txtNumero = new JTextField();
        txtNumero.setBounds(90, 10, 90, 20);
        getContentPane().add(txtNumero);
        txtNumero.setColumns(10);

        btnProcesar = new JButton("Procesar");
        btnProcesar.addActionListener(this);
        btnProcesar.setBounds(335, 9, 89, 23);
        getContentPane().add(btnProcesar);

        btnBorrar = new JButton("Borrar");
        btnBorrar.addActionListener(this);
        btnBorrar.setBounds(246, 9, 89, 23);
        getContentPane().add(btnBorrar);

        scpScroll = new JScrollPane();
        scpScroll.setBounds(10, 44, 414, 75);
        getContentPane().add(scpScroll);

        txtS = new JTextArea();
        txtS.setFont(new Font("Monospaced", Font.PLAIN, 12));
        scpScroll.setViewportView(txtS);
    }

    // Direcciona eventos de tipo ActionEvent
    public void actionPerformed(ActionEvent arg0) {
        if (arg0.getSource() == btnProcesar) {
            actionPerformedBtnProcesar(arg0);
        }
        if (arg0.getSource() == btnBorrar) {
            actionPerformedBtnBorrar(arg0);
        }
    }

```

```

    }

    // Procesa la pulsación del botón Procesar
    protected void actionPerformedBtnProcesar(ActionEvent arg0) {
        // Declaración de variables
        int numero;
        String obsequio = "Ninguno";

        // Entrada de datos
        numero = Integer.parseInt(txtNumero.getText());

        // Determina el obsequio
        if (numero >= 1 && numero <= 20)
            obsequio = "Un lapicero";
        if (numero >= 21 && numero <= 40)
            obsequio = "Un cuaderno de 100 hojas";
        if (numero >= 41 && numero <= 60)
            obsequio = "Una caja de 12 plumones";
        if (numero >= 61 && numero <= 80)
            obsequio = "Un cuaderno espiral";
        if (numero >= 81 && numero <= 99)
            obsequio = "Una agenda";
        if (numero == 100)
            obsequio = "Una mochila";

        // Salida de resultados
        txtS.setText("Obsequio : " + obsequio);
    }

    // Procesa la pulsación del botón Borrar
    protected void actionPerformedBtnBorrar(ActionEvent arg0) {
        txtNumero.setText("");
        txtS.setText("");
        txtNumero.requestFocus();
    }
}

```

### Problema 3

Una dulcería vende chocolates a los precios dados en la siguiente tabla:

| Tipo de chocolate | Precio unitario |
|-------------------|-----------------|
| Primor            | S/. 8.5         |
| Dulzura           | S/. 10.0        |
| Tentación         | S/. 7.0         |
| Explosión         | S/. 12.5        |

Como oferta, la tienda aplica un porcentaje de descuento sobre el importe de la compra, sobre la base de la cantidad de chocolates adquiridos, de acuerdo con la siguiente tabla:

| Cantidad de chocolates | Descuento |
|------------------------|-----------|
| < 5                    | 4.0%      |
| ≥ 5 y < 10             | 6.5%      |
| ≥ 10 y < 15            | 9.0%      |
| ≥ 15                   | 11.5%     |

Adicionalmente, si el importe a pagar es no menor de S/. 250, la tienda obsequia 3 caramelos por cada chocolate; en caso contrario, obsequia 2 caramelos por cada chocolates.

Dado el tipo de chocolate y la cantidad de unidades adquiridas, diseñe un programa que determine el importe de la compra, el importe del descuento, el importe a pagar y la cantidad de caramelos de obsequio.

### Algoritmo

#### Inicio

```
// Declaración de variables
entero tipo, cantidad, caramelos
real impcom = 0, impdes = 0, imppag

// Entrada de datos
Leer tipo, cantidad

// Calcula el importe de la compra
si(tipo == 0)
    impcom = 8.5*cantidad
si(tipo == 1)
    impcom = 10.0*cantidad
si(tipo == 2)
    impcom = 7.0*cantidad
si(tipo == 3)
    impcom = 12.5*cantidad

// Calcula el importe del descuento
si(cantidad < 5)
    impdes = 0.04*impcom
si(cantidad >= 5 && cantidad < 10)
    impdes = 0.065*impcom
si(cantidad >= 10 && cantidad < 15)
    impdes = 0.09*impcom
si(cantidad >= 15)
    impdes = 0.115*impcom

// Calcula el importe a pagar
imppag = impcom - impdes

// Calcula la cantidad de caramelos de regalo
si(imppag < 250)
    caramelos = 2*cantidad
sino
    caramelos = 3*cantidad

// Salida de resultados
Imprimir impcom, impdes, imppag, caramelos
```

Fin

### Programa

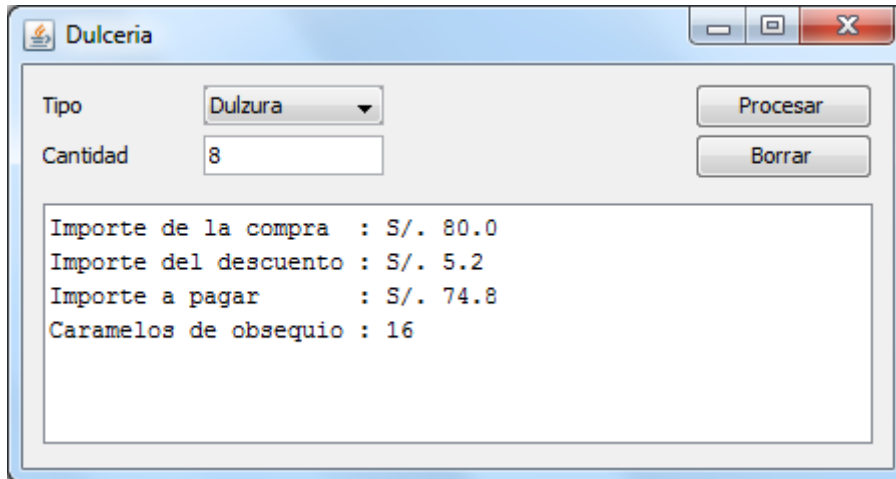
```
package cibertec;

import java.awt.EventQueue;
import java.awt.Font;
```

```

import javax.swing.JFrame;
import javax.swing.JButton;
import javax.swing.UIManager;
import javax.swing.JLabel;
import javax.swing.JTextField;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
import javax.swing.JComboBox;
import javax.swing.DefaultComboBoxModel;

```



```

public class Dulceria extends JFrame implements ActionListener {
    private static final long serialVersionUID = 1L;

    // Declaración de variables
    private JLabel lblTipo;
    private JLabel lblCantidad;
    private JComboBox<String> cboTipo;
    private JTextField txtCantidad;
    private JButton btnProcesar;
    private JButton btnBorrar;
    private JScrollPane scpScroll;
    private JTextArea txtS;

    // Lanza la aplicación
    public static void main(String[] args) {
        try {
            UIManager.setLookAndFeel(
                "com.sun.java.swing.plaf.windows.WindowsLookAndFeel");
        }
        catch (Throwable e) {
            e.printStackTrace();
        }
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    Dulceria frame = new Dulceria();
                    frame.setVisible(true);
                }
                catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }
}

```

```

    }
    });
}

// Crea la GUI
public Dulceria() {
    setTitle("Dulceria");
    setBounds(100, 100, 450, 239);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    getContentPane().setLayout(null);

    lblTipo = new JLabel("Tipo");
    lblTipo.setBounds(10, 13, 80, 14);
    getContentPane().add(lblTipo);

    lblCantidad = new JLabel("Cantidad");
    lblCantidad.setBounds(10, 38, 80, 14);
    getContentPane().add(lblCantidad);

    cboTipo = new JComboBox<String>();
    cboTipo.setModel(new DefaultComboBoxModel<String>(new String[] {
        "Primor", "Dulzura", "Tentación", "Explosión" }));
    cboTipo.setBounds(90, 10, 90, 20);
    getContentPane().add(cboTipo);

    txtCantidad = new JTextField();
    txtCantidad.setBounds(90, 35, 90, 20);
    getContentPane().add(txtCantidad);
    txtCantidad.setColumns(10);

    btnProcesar = new JButton("Procesar");
    btnProcesar.addActionListener(this);
    btnProcesar.setBounds(335, 9, 89, 23);
    getContentPane().add(btnProcesar);

    btnBorrar = new JButton("Borrar");
    btnBorrar.addActionListener(this);
    btnBorrar.setBounds(335, 34, 89, 23);
    getContentPane().add(btnBorrar);

    scpScroll = new JScrollPane();
    scpScroll.setBounds(10, 69, 414, 120);
    getContentPane().add(scpScroll);

    txtS = new JTextArea();
    txtS.setFont(new Font("Monospaced", Font.PLAIN, 12));
    scpScroll.setViewportView(txtS);
}

// Direcciona eventos de tipo ActionEvent
public void actionPerformed(ActionEvent arg0) {
    if (arg0.getSource() == btnProcesar) {
        actionPerformedBtnProcesar(arg0);
    }
    if (arg0.getSource() == btnBorrar) {
        actionPerformedBtnBorrar(arg0);
    }
}
}

```



```

// Procesa la pulsación del botón Procesar
protected void actionPerformedBtnProcesar(ActionEvent arg0) {
    // Declaración de variables
    int tipo, cantidad, caramelos;
    double impcom = 0, impdes = 0, imppag;

    // Entrada de datos
    tipo = cboTipo.getSelectedIndex();
    cantidad = Integer.parseInt(txtCantidad.getText());

    // Calcula el importe de la compra
    if (tipo == 0)
        impcom = 8.5 * cantidad;
    if (tipo == 1)
        impcom = 10.0 * cantidad;
    if (tipo == 2)
        impcom = 7.0 * cantidad;
    if (tipo == 3)
        impcom = 12.5 * cantidad;

    // Calcula el importe del descuento
    if (cantidad < 5)
        impdes = 0.04 * impcom;
    if (cantidad >= 5 && cantidad < 10)
        impdes = 0.065 * impcom;
    if (cantidad >= 10 && cantidad < 15)
        impdes = 0.09 * impcom;
    if (cantidad >= 15)
        impdes = 0.115 * impcom;

    // Calcula el importe a pagar
    imppag = impcom - impdes;

    // Calcula la cantidad de caramelos de regalo
    if (imppag < 250)
        caramelos = 2 * cantidad;
    else
        caramelos = 3 * cantidad;

    // Salida de resultados
    txtS.setText("Importe de la compra : S/. " + impcom + "\n");
    txtS.append("Importe del descuento : S/. " + impdes + "\n");
    txtS.append("Importe a pagar : S/. " + imppag + "\n");
    txtS.append("Caramelos de obsequio : " + caramelos);
}

// Procesa la pulsación del botón Borrar
protected void actionPerformedBtnBorrar(ActionEvent arg0) {
    cboTipo.setSelectedIndex(0);
    txtCantidad.setText("");
    txtS.setText("");
    txtCantidad.requestFocus();
}
}

```

**Problema 4**

Los cálculos salariales de los empleados de una empresa se efectúan de la siguiente manera:

- Sueldo bruto : horas trabajadas x tarifa horaria
- Descuento : 15% del sueldo bruto
- Sueldo neto : sueldo bruto – descuento

La tarifa horaria depende de la categoría del trabajador de acuerdo con la siguiente tabla:

| Categoría | Tarifa horaria (S/.) |
|-----------|----------------------|
| C1        | 45.0                 |
| C2        | 37.5                 |
| C3        | 35.0                 |
| C4        | 32.5                 |

Dadas las horas trabajadas y la categoría de un empleado, diseñe un programa que determine la tarifa horaria, el sueldo bruto, el descuento y el sueldo neto del empleado.

**Algoritmo****Inicio**

```
// Declaración de variables
entero categoria
real horas, suelbru, desc, suelnet, tarifa = 0

// Entrada de datos
Leer categoria, horas

// Determina la tarifa horaria
si(categoria == 0)
    tarifa = 45.0
si(categoria == 1)
    tarifa = 37.5
si(categoria == 2)
    tarifa = 35.0
si(categoria == 3)
    tarifa = 32.5

// Calcula el sueldo bruto
suelbru = horas*tarifa

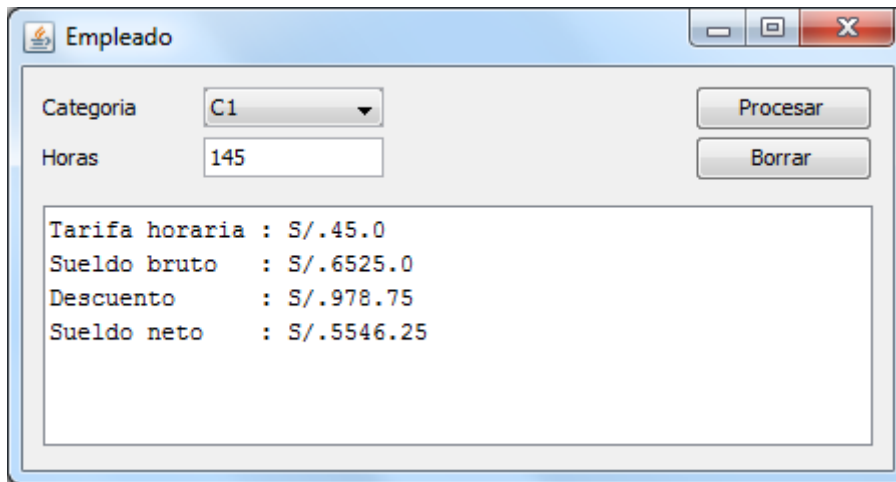
// Calcula el descuento
desc = 0.15*suelbru

// Calcula el sueldo neto
suelnet = suelbru - desc

// Salida de resultados
Imprimir tarifa, suelbru, desc, suelnet
```

**Fin**

## Programa



```
package cibertec;

import java.awt.EventQueue;
import java.awt.Font;
import javax.swing.JFrame;
import javax.swing.JButton;
import javax.swing.UIManager;
import javax.swing.JLabel;
import javax.swing.JTextField;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
import javax.swing.JComboBox;
import javax.swing.DefaultComboBoxModel;

public class Empleado extends JFrame implements ActionListener {
    private static final long serialVersionUID = 1L;

    // Declaración de variables
    private JLabel lblCategoría;
    private JLabel lblHoras;
    private JTextField txtHoras;
    private JButton btnProcesar;
    private JButton btnBorrar;
    private JScrollPane scpScroll;
    private JTextArea txtS;
    private JComboBox<String> cboCategoría;

    // Lanza la aplicación
    public static void main(String[] args) {
        try {
            UIManager.setLookAndFeel(
                "com.sun.java.swing.plaf.windows.WindowsLookAndFeel");
        }
        catch (Throwable e) {
            e.printStackTrace();
        }
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
```

```

        Empleado frame = new Empleado();
        frame.setVisible(true);
    }
    catch (Exception e) {
        e.printStackTrace();
    }
}
});
}

// Crea la GUI
public Empleado() {
    setTitle("Empleado");
    setBounds(100, 100, 450, 239);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    getContentPane().setLayout(null);

    lblCategoría = new JLabel("Categoría");
    lblCategoría.setBounds(10, 13, 80, 14);
    getContentPane().add(lblCategoría);

    lblHoras = new JLabel("Horas");
    lblHoras.setBounds(10, 38, 80, 14);
    getContentPane().add(lblHoras);

    cboCategoría = new JComboBox<String>();
    cboCategoría.setModel(new DefaultComboBoxModel<String>(new String[] {
        "C1", "C2", "C3", "C4" }));
    cboCategoría.setBounds(90, 10, 90, 20);
    getContentPane().add(cboCategoría);

    txtHoras = new JTextField();
    txtHoras.setBounds(90, 35, 90, 20);
    getContentPane().add(txtHoras);
    txtHoras.setColumns(10);

    btnProcesar = new JButton("Procesar");
    btnProcesar.addActionListener(this);
    btnProcesar.setBounds(335, 9, 89, 23);
    getContentPane().add(btnProcesar);

    btnBorrar = new JButton("Borrar");
    btnBorrar.addActionListener(this);
    btnBorrar.setBounds(335, 34, 89, 23);
    getContentPane().add(btnBorrar);

    scpScroll = new JScrollPane();
    scpScroll.setBounds(10, 69, 414, 120);
    getContentPane().add(scpScroll);

    txtS = new JTextArea();
    txtS.setFont(new Font("Monospaced", Font.PLAIN, 12));
    scpScroll.setViewportView(txtS);
}

// Direcciona eventos de tipo ActionEvent
public void actionPerformed(ActionEvent arg0) {
    if (arg0.getSource() == btnProcesar) {
        actionPerformedBtnProcesar(arg0);
    }
}

```

```

    }
    if (arg0.getSource() == btnBorrar) {
        actionPerformedBtnBorrar(arg0);
    }
}

// Procesa la pulsación del botón Procesar
protected void actionPerformedBtnProcesar(ActionEvent arg0) {
    // Declaración de variables
    int categoria;
    double horas, suelbru, desc, suelnet, tarifa = 0;

    // Entrada de datos
    categoria = cboCategoria.getSelectedIndex();
    horas = Double.parseDouble(txtHoras.getText());

    // Determina la tarifa horaria
    if (categoria == 0)
        tarifa = 45.0;
    if (categoria == 1)
        tarifa = 37.5;
    if (categoria == 2)
        tarifa = 35.0;
    if (categoria == 3)
        tarifa = 32.5;

    // Calcula el sueldo bruto
    suelbru = horas * tarifa;

    // Calcula el descuento
    desc = 0.15 * suelbru;

    // Calcula el sueldo neto
    suelnet = suelbru - desc;

    // Salida de resultados
    txtS.setText("Tarifa horaria : S/." + tarifa + "\n");
    txtS.append("Sueldo bruto : S/." + suelbru + "\n");
    txtS.append("Descuento : S/." + desc + "\n");
    txtS.append("Sueldo neto : S/." + suelnet);
}

// Procesa la pulsación del botón Borrar
protected void actionPerformedBtnBorrar(ActionEvent arg0) {
    cboCategoria.setSelectedIndex(0);
    txtHoras.setText("");
    txtS.setText("");
    txtHoras.requestFocus();
}
}

```

**Problema 5**

Los ángulos se clasifican de la siguiente manera:

| Magnitud                        | Clasificación |
|---------------------------------|---------------|
| $\beta = 0^\circ$               | Nulo          |
| $0^\circ < \beta < 90^\circ$    | Agudo         |
| $\beta = 90^\circ$              | Recto         |
| $90^\circ < \beta < 180^\circ$  | Obtuso        |
| $\beta = 180^\circ$             | Llano         |
| $180^\circ < \beta < 360^\circ$ | Cóncavo       |
| $\beta = 360^\circ$             | Completo      |

Diseñe un algoritmo que determine la clasificación de un ángulo dado en grados, minutos y segundos. Asuma que el ángulo está en el intervalo de  $0^\circ$  a  $360^\circ$ .

**Algoritmo****Inicio**

```
// Declaración de variables
entero grados, minutos, segundos
real beta
cadena tipo = ""

// Entrada de datos
Leer grados, minutos, segundos

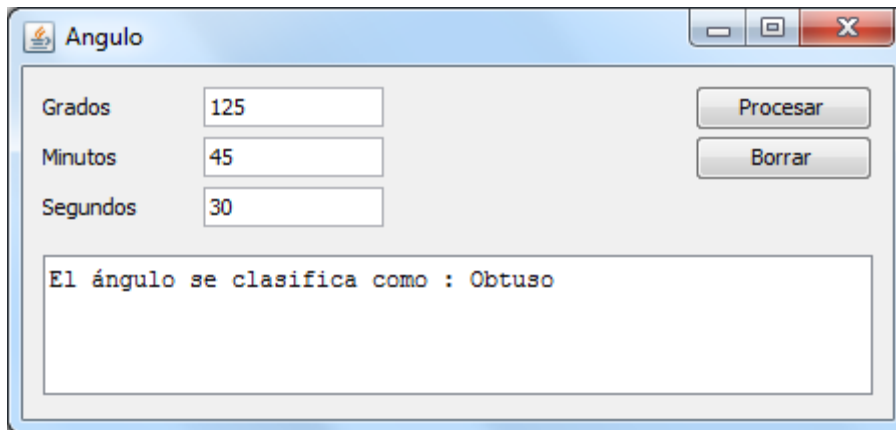
// Determina el ángulo en grados
beta = grados + minutos/60.0 + segundos/3600.0

// Determina el tipo de ángulo
si( beta == 0 )
    tipo = "Nulo"
si( beta > 0 && beta < 90 )
    tipo = "Agudo"
si( beta == 90 )
    tipo = "Recto"
si( beta > 90 && beta < 180 )
    tipo = "Obtuso"
si( beta == 180 )
    tipo = "Llano"
si( beta > 180 && beta < 360 )
    tipo = "Cóncavo"
si( beta == 360 )
    tipo = "Completo"

// Salida de resultados
Imprimir tipo
```

**Fin**

## Programa



```
package cibertec;

import java.awt.EventQueue;
import javax.swing.JFrame;
import javax.swing.JButton;
import javax.swing.UIManager;
import javax.swing.JLabel;
import javax.swing.JTextField;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
import java.awt.Font;

public class Angulo extends JFrame implements ActionListener {
    private static final long serialVersionUID = 1L;

    // Declaración de variables
    private JLabel lblGrados;
    private JLabel lblMinutos;
    private JLabel lblSegundos;
    private JTextField txtGrados;
    private JTextField txtMinutos;
    private JTextField txtSegundos;
    private JButton btnProcesar;
    private JButton btnBorrar;
    private JScrollPane scpScroll;
    private JTextArea txtS;

    // Lanza la aplicación
    public static void main(String[] args) {
        try {
            UIManager.setLookAndFeel(
                "com.sun.java.swing.plaf.windows.WindowsLookAndFeel");
        }
        catch (Throwable e) {
            e.printStackTrace();
        }
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    Angulo frame = new Angulo();

```

```

        frame.setVisible(true);
    }
    catch (Exception e) {
        e.printStackTrace();
    }
}
});
}

// Crea la GUI
public Angulo() {
    setTitle("Angulo");
    setBounds(100, 100, 450, 214);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    getContentPane().setLayout(null);

    lblGrados = new JLabel("Grados");
    lblGrados.setBounds(10, 13, 80, 14);
    getContentPane().add(lblGrados);

    lblMinutos = new JLabel("Minutos");
    lblMinutos.setBounds(10, 38, 80, 14);
    getContentPane().add(lblMinutos);

    lblSegundos = new JLabel("Segundos");
    lblSegundos.setBounds(10, 63, 80, 14);
    getContentPane().add(lblSegundos);

    txtGrados = new JTextField();
    txtGrados.setBounds(90, 10, 90, 20);
    getContentPane().add(txtGrados);
    txtGrados.setColumns(10);

    txtMinutos = new JTextField();
    txtMinutos.setBounds(90, 35, 90, 20);
    getContentPane().add(txtMinutos);
    txtMinutos.setColumns(10);

    txtSegundos = new JTextField();
    txtSegundos.setBounds(90, 60, 90, 20);
    getContentPane().add(txtSegundos);
    txtSegundos.setColumns(10);

    btnProcesar = new JButton("Procesar");
    btnProcesar.addActionListener(this);
    btnProcesar.setBounds(335, 9, 89, 23);
    getContentPane().add(btnProcesar);

    btnBorrar = new JButton("Borrar");
    btnBorrar.addActionListener(this);
    btnBorrar.setBounds(335, 34, 89, 23);
    getContentPane().add(btnBorrar);

    scpScroll = new JScrollPane();
    scpScroll.setBounds(10, 94, 414, 70);
    getContentPane().add(scpScroll);

    txtS = new JTextArea();
    txtS.setFont(new Font("Monospaced", Font.PLAIN, 12));

```



```

        scpScroll.setViewportView(txtS);
    }

    // Direcciona eventos de tipo ActionEvent
    public void actionPerformed(ActionEvent arg0) {
        if (arg0.getSource() == btnProcesar) {
            actionPerformedBtnProcesar(arg0);
        }
        if (arg0.getSource() == btnBorrar) {
            actionPerformedBtnBorrar(arg0);
        }
    }

    // Procesa la pulsación del botón Procesar
    protected void actionPerformedBtnProcesar(ActionEvent arg0) {
        // Declaración de variables
        int grados, minutos, segundos;
        double beta;
        String tipo = "";

        // Entrada de datos
        grados = Integer.parseInt(txtGrados.getText());
        minutos = Integer.parseInt(txtMinutos.getText());
        segundos = Integer.parseInt(txtSegundos.getText());

        // Determina el ángulo en grados
        beta = grados + minutos / 60.0 + segundos / 3600.0;

        // Determina el tipo de ángulo
        if (beta == 0)
            tipo = "Nulo";
        if (beta > 0 && beta < 90)
            tipo = "Agudo";
        if (beta == 90)
            tipo = "Recto";
        if (beta > 90 && beta < 180)
            tipo = "Obtuso";
        if (beta == 180)
            tipo = "Llano";
        if (beta > 180 && beta < 360)
            tipo = "Cóncavo";
        if (beta == 360)
            tipo = "Completo";

        // Salida de resultados
        txtS.setText("El ángulo se clasifica como : " + tipo);
    }

    // Procesa la pulsación del botón Borrar
    protected void actionPerformedBtnBorrar(ActionEvent arg0) {
        txtGrados.setText("");
        txtMinutos.setText("");
        txtSegundos.setText("");
        txtS.setText("");
        txtGrados.requestFocus();
    }
}

```

## Problema 6

El promedio final de un curso se obtiene sobre la base del promedio simple de tres prácticas calificadas. Para ayudar a los alumnos, el profesor del curso ha decidido incrementar en dos puntos las notas de las prácticas no menores que 10. Dadas las tres notas de práctica de un estudiante, diseñe un programa que determine el promedio final que le corresponde. Considere que la nota máxima es 20.

### Algoritmo

#### Inicio

```
// Declaración de variables
real p1, p2, p3, promedio

// Entrada de datos
Leer p1, p2, p3

// Si amerita, añade 2 puntos a p1
si( p1 >= 10 ){
    p1 = p1 + 2
    si( p1 > 20 )
        p1 = 20
}

// Si amerita, añade 2 puntos a p2
si( p2 >= 10 ){
    p2 = p2 + 2
    si( p2 > 20 )
        p2 = 20
}

// Si amerita, añade 2 puntos a p3
si( p3 >= 10 ){
    p3 = p3 + 2
    si( p3 > 20 )
        p3 = 20
}

// Cálculo del promedio
promedio = (p1+p2+p3)/3

// Salida de resultados
Imprimir promedio
```

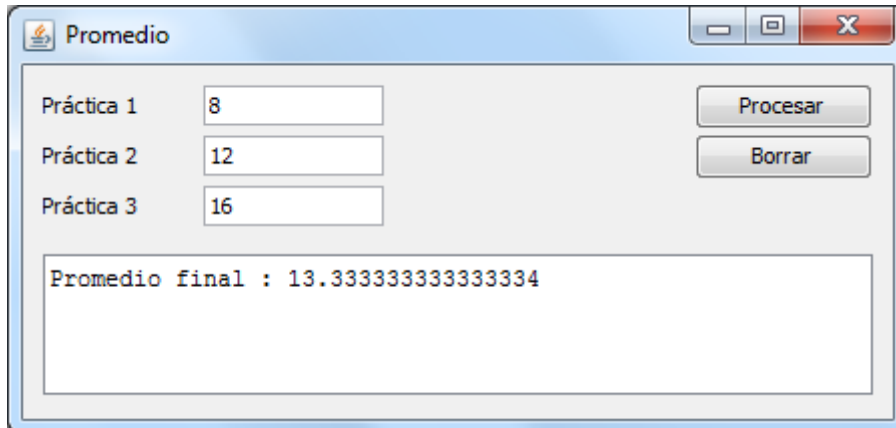
#### Fin

### Programa

```
package cibertec;

import java.awt.EventQueue;
import java.awt.Font;
import javax.swing.JFrame;
import javax.swing.JButton;
import javax.swing.UIManager;
import javax.swing.JLabel;
import javax.swing.JTextField;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
```

```
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
```



```
public class Promedio extends JFrame implements ActionListener {
    private static final long serialVersionUID = 1L;

    // Declaración de variables
    private JLabel lblPractica1;
    private JLabel lblPractica2;
    private JLabel lblPractica3;
    private JTextField txtPractica1;
    private JTextField txtPractica2;
    private JTextField txtPractica3;
    private JButton btnProcesar;
    private JButton btnBorrar;
    private JScrollPane scpScroll;
    private JTextArea txtS;

    // Lanza la aplicación
    public static void main(String[] args) {
        try {
            UIManager.setLookAndFeel(
                "com.sun.java.swing.plaf.windows.WindowsLookAndFeel");
        }
        catch (Throwable e) {
            e.printStackTrace();
        }
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    Promedio frame = new Promedio();
                    frame.setVisible(true);
                }
                catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    // Crea la GUI
    public Promedio() {
        setTitle("Promedio");
        setBounds(100, 100, 450, 214);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
```

```

getContentPane().setLayout(null);

lblPractica1 = new JLabel("Práctica 1");
lblPractica1.setBounds(10, 13, 80, 14);
getContentPane().add(lblPractica1);

lblPractica2 = new JLabel("Práctica 2");
lblPractica2.setBounds(10, 38, 80, 14);
getContentPane().add(lblPractica2);

lblPractica3 = new JLabel("Práctica 3");
lblPractica3.setBounds(10, 63, 80, 14);
getContentPane().add(lblPractica3);

txtPractica1 = new JTextField();
txtPractica1.setBounds(90, 10, 90, 20);
getContentPane().add(txtPractica1);
txtPractica1.setColumns(10);

txtPractica2 = new JTextField();
txtPractica2.setBounds(90, 35, 90, 20);
getContentPane().add(txtPractica2);
txtPractica2.setColumns(10);

txtPractica3 = new JTextField();
txtPractica3.setBounds(90, 60, 90, 20);
getContentPane().add(txtPractica3);
txtPractica3.setColumns(10);

btnProcesar = new JButton("Procesar");
btnProcesar.addActionListener(this);
btnProcesar.setBounds(335, 9, 89, 23);
getContentPane().add(btnProcesar);

btnBorrar = new JButton("Borrar");
btnBorrar.addActionListener(this);
btnBorrar.setBounds(335, 34, 89, 23);
getContentPane().add(btnBorrar);

scpScroll = new JScrollPane();
scpScroll.setBounds(10, 94, 414, 70);
getContentPane().add(scpScroll);

txtS = new JTextArea();
txtS.setFont(new Font("Monospaced", Font.PLAIN, 12));
scpScroll.setViewportView(txtS);
}

// Direcciona eventos de tipo ActionEvent
public void actionPerformed(ActionEvent arg0) {
    if (arg0.getSource() == btnProcesar) {
        actionPerformedBtnProcesar(arg0);
    }
    if (arg0.getSource() == btnBorrar) {
        actionPerformedBtnBorrar(arg0);
    }
}

// Procesa la pulsación del botón Procesar

```

```

protected void actionPerformedBtnProcesar(ActionEvent arg0) {
    // Declaración de variables
    double promedio;
    int p1, p2, p3;

    // Entrada de datos
    p1 = Integer.parseInt(txtPractica1.getText());
    p2 = Integer.parseInt(txtPractica2.getText());
    p3 = Integer.parseInt(txtPractica3.getText());

    // Si amerita, añade 2 puntos a p1
    if (p1 >= 10) {
        p1 = p1 + 2;
        if (p1 > 20)
            p1 = 20;
    }

    // Si amerita, añade 2 puntos a p2
    if (p2 >= 10) {
        p2 = p2 + 2;
        if (p2 > 20)
            p2 = 20;
    }

    // Si amerita, añade 2 puntos a p3
    if (p3 >= 10) {
        p3 = p3 + 2;
        if (p3 > 20)
            p3 = 20;
    }

    // Calcula el promedio
    promedio = (p1 + p2 + p3) / 3.0;

    // Salida de resultados
    txtS.setText("Promedio final : " + promedio);
}

// Procesa la pulsación del botón Borrar
protected void actionPerformedBtnBorrar(ActionEvent arg0) {
    txtPractica1.setText("");
    txtPractica2.setText("");
    txtPractica3.setText("");
    txtS.setText("");
    txtPractica1.requestFocus();
}
}

```

## Problema 7

Dadas las edades de tres personas, diseñe un programa que determine la edad mayor.

### Algoritmo

#### Inicio

```

// Declaración de variables
entero edad1, edad2, edad3, edadMayor

```

```

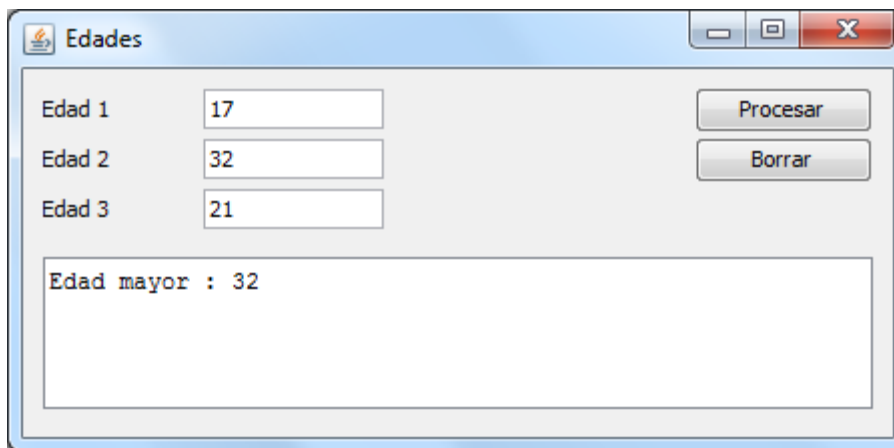
// Entrada de datos
Leer edad1, edad2, edad3

// Determina la edad mayor
edadMayor = edad1
si (edad2 > edadMayor)
    edadMayor = edad2
si (edad3 > edadMayor)
    edadMayor = edad3

// Salida de resultados
Imprimir edadMayor
Fin

```

## Programa



```

package cibertec;

import java.awt.EventQueue;
import java.awt.Font;
import javax.swing.JFrame;
import javax.swing.JButton;
import javax.swing.UIManager;
import javax.swing.JLabel;
import javax.swing.JTextField;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;

public class Edades extends JFrame implements ActionListener {
    private static final long serialVersionUID = 1L;

    // Declaración de variables
    private JLabel lblEdad1;
    private JLabel lblEdad2;
    private JLabel lblEdad3;
    private JTextField txtEdad1;
    private JTextField txtEdad2;
    private JTextField txtEdad3;
    private JButton btnProcesar;
    private JButton btnBorrar;
    private JScrollPane scpScroll;
    private JTextArea txtS;

```

```

// Lanza la aplicación
public static void main(String[] args) {
    try {
        UIManager.setLookAndFeel(
            "com.sun.java.swing.plaf.windows.WindowsLookAndFeel");
    }
    catch (Throwable e) {
        e.printStackTrace();
    }
    EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                Edades frame = new Edades();
                frame.setVisible(true);
            }
            catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}

// Crea la GUI
public Edades() {
    setTitle("Edades");
    setBounds(100, 100, 450, 222);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    getContentPane().setLayout(null);

    lblEdad1 = new JLabel("Edad 1");
    lblEdad1.setBounds(10, 13, 80, 14);
    getContentPane().add(lblEdad1);

    lblEdad2 = new JLabel("Edad 2");
    lblEdad2.setBounds(10, 38, 80, 14);
    getContentPane().add(lblEdad2);

    lblEdad3 = new JLabel("Edad 3");
    lblEdad3.setBounds(10, 63, 80, 14);
    getContentPane().add(lblEdad3);

    txtEdad1 = new JTextField();
    txtEdad1.setBounds(90, 10, 90, 20);
    getContentPane().add(txtEdad1);
    txtEdad1.setColumns(10);

    txtEdad2 = new JTextField();
    txtEdad2.setBounds(90, 35, 90, 20);
    getContentPane().add(txtEdad2);
    txtEdad2.setColumns(10);

    txtEdad3 = new JTextField();
    txtEdad3.setBounds(90, 60, 90, 20);
    getContentPane().add(txtEdad3);
    txtEdad3.setColumns(10);

    btnProcesar = new JButton("Procesar");
    btnProcesar.addActionListener(this);
}

```

```

        btnProcesar.setBounds(335, 9, 89, 23);
        getContentPane().add(btnProcesar);

        btnBorrar = new JButton("Borrar");
        btnBorrar.addActionListener(this);
        btnBorrar.setBounds(335, 34, 89, 23);
        getContentPane().add(btnBorrar);

        scpScroll = new JScrollPane();
        scpScroll.setBounds(10, 94, 414, 76);
        getContentPane().add(scpScroll);

        txtS = new JTextArea();
        txtS.setFont(new Font("Monospaced", Font.PLAIN, 12));
        scpScroll.setViewportView(txtS);
    }

    // Direcciona eventos de tipo ActionEvent
    public void actionPerformed(ActionEvent arg0) {
        if (arg0.getSource() == btnProcesar) {
            actionPerformedBtnProcesar(arg0);
        }
        if (arg0.getSource() == btnBorrar) {
            actionPerformedBtnBorrar(arg0);
        }
    }

    // Procesa la pulsación del botón Procesar
    protected void actionPerformedBtnProcesar(ActionEvent arg0) {
        // Declaración de variables
        int edad1, edad2, edad3, edadMayor;

        // Entrada de datos
        edad1 = Integer.parseInt(txtEdad1.getText());
        edad2 = Integer.parseInt(txtEdad2.getText());
        edad3 = Integer.parseInt(txtEdad3.getText());

        // Determina la edad mayor
        edadMayor = edad1;
        if (edad2 > edadMayor)
            edadMayor = edad2;
        if (edad3 > edadMayor)
            edadMayor = edad3;

        // Salida de resultados
        txtS.setText("Edad mayor : " + edadMayor);
    }

    // Procesa la pulsación del botón Borrar
    protected void actionPerformedBtnBorrar(ActionEvent arg0) {
        txtEdad1.setText("");
        txtEdad2.setText("");
        txtEdad3.setText("");
        txtS.setText("");
        txtEdad1.requestFocus();
    }
}

```