

# Conceptos Básicos de Algoritmia

---

## LOGRO DE LA UNIDAD DE APRENDIZAJE

Al finalizar la unidad, el alumno define las etapas de desarrollo de un algoritmo y diseña algoritmos básicos en pseudocódigo

## TEMARIO

1. Algoritmo
2. Clasificación de los algoritmos
3. Programa
4. Etapas de un algoritmo
5. Variable
6. Pseudocódigo
7. Instrucciones algorítmicas básicas
8. Etapas de un algoritmo computacional
9. Operadores aritméticos

## ACTIVIDADES

Los alumnos desarrollan algoritmos básicos con fórmulas definidas.

## 1. Algoritmo

Un algoritmo es un método de solución de un problema expresado a través de un conjunto de pasos, procedimientos o acciones. Son algoritmos los siguientes:

- Las instrucciones para instalar un equipo de sonido
- Una receta para preparar un plato de comida
- Las instrucciones para hallar el máximo común divisor de dos números
- Las instrucciones para convertir una cantidad dada en soles a dólares

### Ejemplo 1

El siguiente algoritmo determina el índice de masa corporal de una persona (IMC)

```
Inicio
    Obtener peso y estatura
    Calcular:  $IMC = \text{peso} / (\text{estatura} * \text{estatura})$ 
    Mostrar IMC
Fin
```

### Ejemplo 2

El siguiente algoritmo determine el área y el precio de un terreno rectangular cuyo costo por metro cuadrado es S/. 750

```
Inicio
    Obtener largo y ancho
    Calcular:  $\text{areaterreno} = \text{largo} * \text{ancho}$ 
    Calcular:  $\text{precioterreno} = \text{areaterreno} * 750$ 
    Mostrar areaterreno y precioterreno
Fin
```

## 2. Clasificación de los algoritmos

Los algoritmos pueden clasificarse en *algoritmos computacionales* y *algoritmos no computacionales*.

Un algoritmo es no computacional cuando no puede ser ejecutado en una computadora. El ejecutor de este tipo de algoritmo es un ser humano. Por ejemplo:

- Las instrucciones para instalar un equipo de sonido
- Una receta para preparar un plato de comida

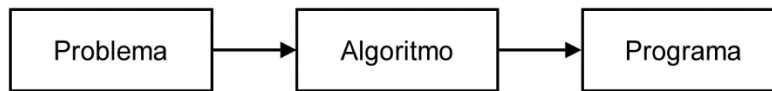
Un algoritmo es computacional cuando puede ser ejecutado en una computadora. Por ejemplo:

- Las instrucciones para hallar el máximo común divisor de dos números
- Las instrucciones para convertir una cantidad dada en soles a dólares

### 3. Programa

Un programa es un conjunto de instrucciones expresadas mediante un lenguaje de programación como Java, C, C++, etc. Las instrucciones del programa se obtienen escribiendo las instrucciones del algoritmo mediante el lenguaje de programación elegido. A las instrucciones de un programa expresadas mediante un lenguaje de programación se denomina también *código fuente*.

Un programa surge ante la necesidad de automatizar la solución de un problema. La Figura 1 muestra que antes de la escritura de un programa, se necesita diseñar un algoritmo. De esta manera, podemos decir, que un programa es la implementación de un algoritmo mediante un lenguaje de programación de forma que sea entendible por el procesador.



**Figura 1** Problema, Algoritmo y Programa

### 4. Etapas de un algoritmo

Todo algoritmo tiene tres etapas claramente diferenciadas: entrada de datos, proceso de cálculo y salida de resultados.

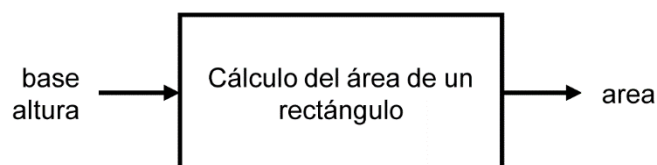
Generalmente, como es el caso de este curso, los datos de entrada se toman de la pantalla del programa y los datos de salida se muestran en la pantalla del mismo programa.



**Figura 2** Etapas de un algoritmo

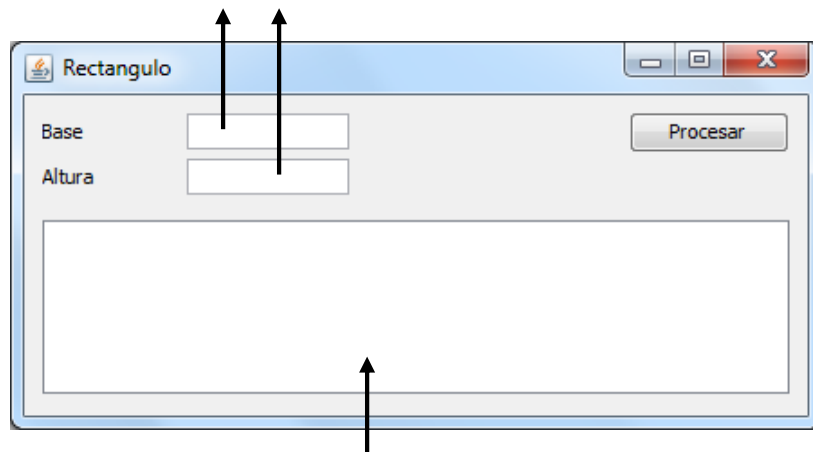
#### Ejemplo 3

En el siguiente gráfico, se especifican los datos de entrada y de salida para el cálculo del área de un rectángulo.



En la figura que sigue, se muestra de dónde se toman los datos de entrada y dónde se ponen los datos de salida.

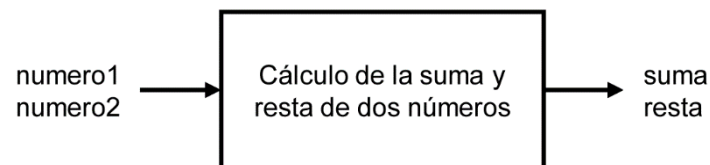
De aquí se toman los datos de entrada



Aquí se ponen los datos de salida

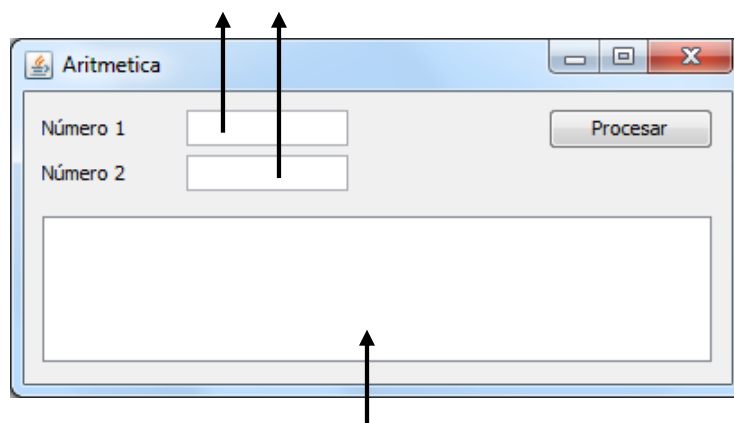
#### Ejemplo 4

En el siguiente gráfico, se especifican los datos de entrada y de salida para el cálculo de la suma y la resta de dos números.



En la figura que sigue, se muestra de dónde se toman los datos de entrada y dónde se ponen los datos de salida.

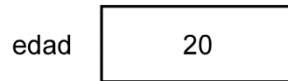
De aquí se toman los datos de entrada



Aquí se ponen los datos de salida

## 5. Variable

Una variable es el nombre asignado a una zona (casilla) de la memoria RAM durante la ejecución de un programa con la finalidad de almacenar un dato o valor. En la Figura 3, se muestra una variable llamada **edad** que almacena un valor **entero** igual a 20.



**Figura 3** Variable

Todas las variables deben ser declaradas antes de ser utilizadas. Declarar una variable consiste en especificar su **nombre** y el **tipo de dato** que almacenará, de acuerdo con una de las siguientes reglas de declaración:

Para una sola variable:

**tipo nombre;**

Para varias variables del mismo tipo:

**tipo nombre1, nombre2, ...,nombren;**

En el nombre de una variable deben tenerse en cuenta las siguientes reglas:

- Debe comenzar con una letra, un símbolo de subrayado (\_) o un símbolo de dólar (\$). Los siguientes caracteres pueden ser letras, dígitos, símbolos de subrayado (\_) o símbolos de dólar (\$).
- No puede ser una palabra reservada del lenguaje.
- Las mayúsculas y minúsculas se consideran diferentes

El tipo de dato de la variable puede ser uno de la siguiente tabla:

**Tabla 1** Tipos de datos básicos del lenguaje Java

Tipo de dato	Naturaleza del dato
Int	Número entero
Double	Número decimal
String	Conjunto de caracteres

### Ejemplo 5

La siguiente instrucción declara la variable **numhij** para almacenar el número de hijos de una persona. El número de hijos es un valor entero, por lo que su tipo de dato es **int**.

```
int numhij;
```

Esto crea la siguiente casilla de memoria sin contenido.

numhij

### Ejemplo 6

La siguiente instrucción declara la variable **peso** para almacenar el peso de una persona. El peso es un valor puede ser decimal, por lo que su tipo de dato es **double**.

```
double peso;
```

Esto crea la siguiente casilla de memoria sin contenido.

peso

### Ejemplo 7

La siguiente instrucción declara la variable **nomape** para almacenar los nombres y apellidos de una persona. Los nombres y apellidos de una persona están formados por un conjunto de caracteres, por lo que el tipo de dato es **String**.

```
String nomape;
```

Esto crea la siguiente casilla de memoria sin contenido.

nomape

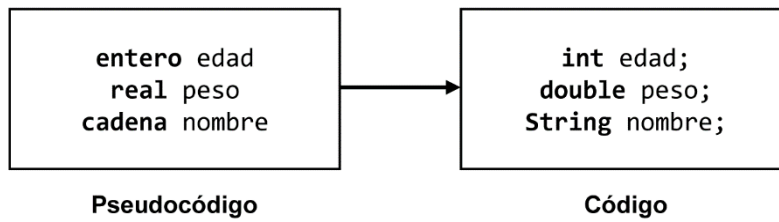
## 6. Pseudocódigo

A las instrucciones de un programa escritas en un lenguaje de programación se denomina *código fuente*. Antes de escribir el *código fuente*, el programador escribe un *pseudocódigo*, esto es, una *imitación del código*. Como imitación del código, el pseudocódigo no tiene reglas formales, varía de un programador a otro.

Un pseudocódigo puede contener símbolos (+, -, \*, /, =, etc.), términos (leer, imprimir, abrir, cerrar, etc.) y estructuras de programación (si, si...sino, hacer...mientras, mientras...hacer, para...mientras).

La ventaja del pseudocódigo es que, su uso, en la planificación de un programa, permite al programador concentrarse en la lógica y en las estructuras de control y no preocuparse de las reglas de un lenguaje específico.

En la Figura 4, puede verse la relación entre pseudocódigo y código. Por ejemplo, en el código se usa el tipo **int**, que corresponde al lenguaje Java, en el pseudocódigo se usa el tipo **entero**.



**Figura 4** Pseudocódigo y Código

## 7. Instrucciones algorítmicas básicas

### 7.1 Declaración de variables

Para declarar variables en el algoritmo, imitaremos la forma de declarar variables del lenguaje Java usando los *tipos de datos algorítmicos* mostrados en la siguiente tabla:

**Tabla 2** Tipos de datos algorítmicos

Java	Algoritmo
int	entero
double	real
char	caracter
String	cadena
boolean	logico

### 7.2 Entrada de datos

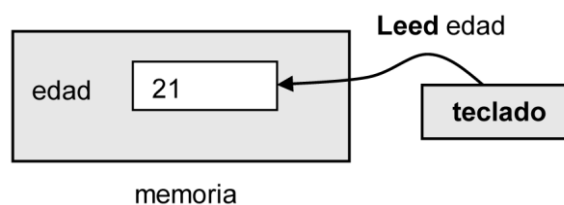
La entrada consiste en obtener un dato desde algún dispositivo de entrada y trasladarlo a la memoria para ser almacenada en una variable (Figura 5). En general, la entrada de una variable se escribe en el pseudocódigo de la siguiente forma:

**Leer variable**

Por ejemplo:

**Leer edad**

Ingresa una valor para las variable edad.



**Figura 5** Entrada

### 7.3 Salida de resultados

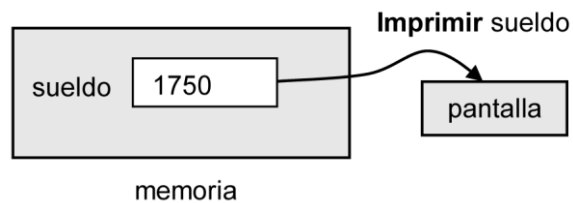
La salida consiste en trasladar a algún dispositivo de salida el valor de una variable (Figura 6). En general, la salida de una variable a la pantalla se escribe en el pseudocódigo de la siguiente forma:

**Imprimir** variable

Por ejemplo:

**Imprimir** sueldo

Imprime el valor de la variable sueldo.



**Figura 6** Salida

### 7.4 Asignación

La asignación consiste en almacenar un valor en una variable. La forma general de asignación es la siguiente:

**variable = expresión**

El símbolo = se denomina *operador de asignación simple*.

Por ejemplo:

area = b\*h

Esto almacena en la variable area el resultado de multiplicar b por h.

## 8. Etapas de desarrollo de un algoritmo computacional

Todo algoritmo computacional puede dividirse en cuatro etapas:

- Declaración de variables
- Entrada de datos
- Proceso de cálculo
- Salida de resultados

### Ejemplo 8

A continuación, se muestra el pseudocódigo del algoritmo del Ejemplo 1 incluyendo todas las etapas de desarrollo de un algoritmo computacional.

**Inicio**



```

// Declaración de variables
real peso, estatura, IMC

// Entrada de datos
Leer peso, estatura

// Proceso de cálculo
IMC = peso/(estatura*estatura)

// Salida de resultados
Imprimir IMC
Fin

```

### Ejemplo 9

A continuación, se muestra el pseudocódigo del algoritmo del Ejemplo 2 incluyendo todas las etapas de desarrollo de un algoritmo computacional.

```

Inicio
// Declaración de variables
real largo, ancho, areaterreno, precioterreno

// Entrada de datos
Leer largo, ancho

// Proceso de cálculo
areaterreno = largo*ancho
precioterreno = areaterreno*750

// Salida de resultados
Imprimir areaterreno, precioterreno
Fin

```

### Ejemplo 10

Diseñe un algoritmo que determine el área de un círculo y la longitud de su circunferencia conociendo su radio. Considere las siguientes fórmulas:

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

$$C = 2 \times 3.1416 \times r$$

Siendo  $A$  el área y  $C$  la longitud de la circunferencia.

### Solución

```

Inicio
// Declaración de variables
real r, A, C

// Entrada de datos
Leer r

// Proceso de cálculo
A = 3.1416*r*r
C = 3.1416*r

// Salida de resultados

```

Imprimir A, C  
Fin

## 9. Operadores aritméticos

En la tabla que sigue, se muestran los operadores aritméticos del lenguaje Java.

**Tabla 3** Operadores aritméticos

Operador	Significado	Ejemplo	Resultado
-	Resta	a-b	Resta de a y b
+	Suma	a+b	Suma de a y b
*	Multiplicación	a*b	Producto de a por b
/	División	a/b	Cociente de a entre b
%	Residuo	a%b	Residuo de a entre b

Los operadores aritméticos pueden utilizarse con tipos enteros y reales. Si ambos operandos son enteros, el resultado es un entero; si alguno de ellos es real, el resultado es real.

### Ejemplo 11

2 + 5	produce el valor	7
2.0 + 5	produce el valor	7.0
2 + 5.0	produce el valor	7.0
2.0 + 5.0	produce el valor	7.0
10/4	produce el valor	2
10/4.0	produce el valor	2.5
10.0/4	produce el valor	2.5
10.0/4.0	produce el valor	2.5
15/2	produce el valor	7
15%2	produce el valor	1
4/10	produce el valor	0
4%10	produce el valor	4

# Problemas propuestos

## Actividades

1. Diseñe un algoritmo que determine el área ( $A$ ) y el perímetro ( $P$ ) de un rectángulo del que se conoce su base ( $b$ ) y su altura ( $h$ ). Considere las siguientes fórmulas:

$$A = b \times h$$
$$P = 2 \times (b + h)$$

2. Diseñe un algoritmo que determine el área ( $A$ ) de un rombo del que se conoce su base mayor ( $B$ ), su base menor ( $b$ ) y su altura ( $h$ ). Considere la siguiente fórmula:

$$A = \frac{(b + B) \times h}{2}$$

3. Diseñe un algoritmo que determine el área ( $A$ ) de un rombo del que se conoce su diagonal mayor ( $D$ ) y su diagonal menor ( $d$ ). Considere la siguiente fórmula:

$$A = \frac{D \times d}{2}$$

## Autoevaluación

1. Diseñe un algoritmo que determine la frecuencia cardiaca de un varón conociendo su edad en años y su peso en kilogramos de acuerdo con la siguiente fórmula:

$$frecuencia = 210 - (0.5 \times edad) - (0.01 \times peso + 4)$$

2. Diseñe un algoritmo que determine el área ( $A$ ) y el volumen ( $V$ ) de un cubo del que se conoce su lado ( $l$ ). Considere las siguientes fórmulas:

$$A = 6l^2$$
$$V = l^3$$

## Para recordar

- Un algoritmo es un método para resolver un problema expresado mediante una serie de pasos.
- El código fuente de un programa se obtiene expresando las instrucciones de un algoritmo mediante un lenguaje de programación.
- El pseudocódigo es la expresión del algoritmo mediante un lenguaje informal de pseudoprogramación que es una imitación de uno a más lenguajes de programación.
- Una variable es el nombre de una localización o casillero de memoria RAM en el que se puede guardar un dato.
- Antes de usar una variable, primero hay que declararla.
- Declarar una variable consiste en especificar su tipo de dato y su nombre.
- Una sentencia de asignación es una instrucción que almacena un valor en la memoria RAM.
- Los operadores aritméticos pueden utilizarse con tipos enteros y reales. Si ambos operandos son enteros, el resultado es un entero; si alguno de ellos es real, el resultado es real.

## **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.

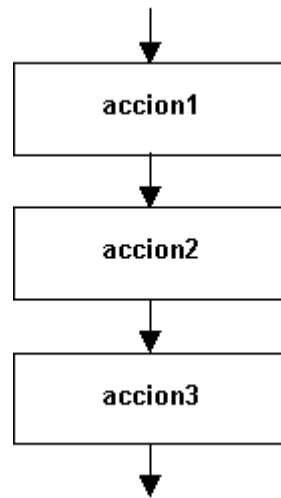


Figura 1 Estructura de Secuencia

## 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

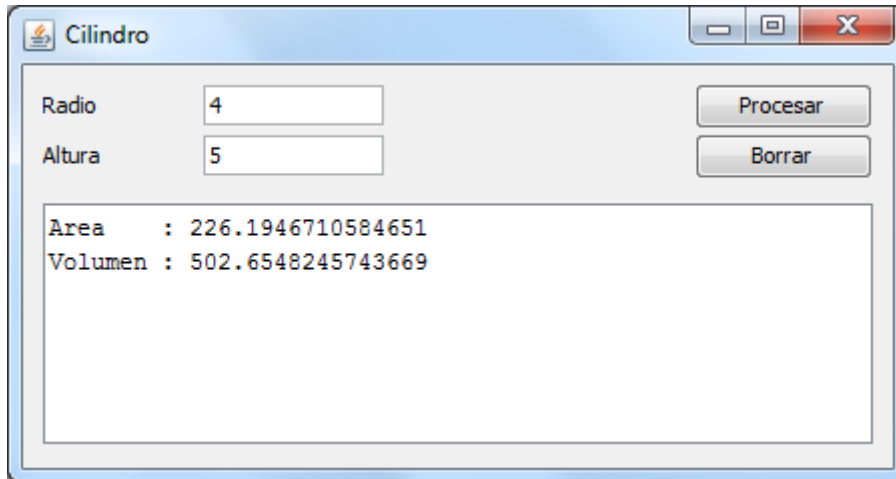
### Programa

```
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;

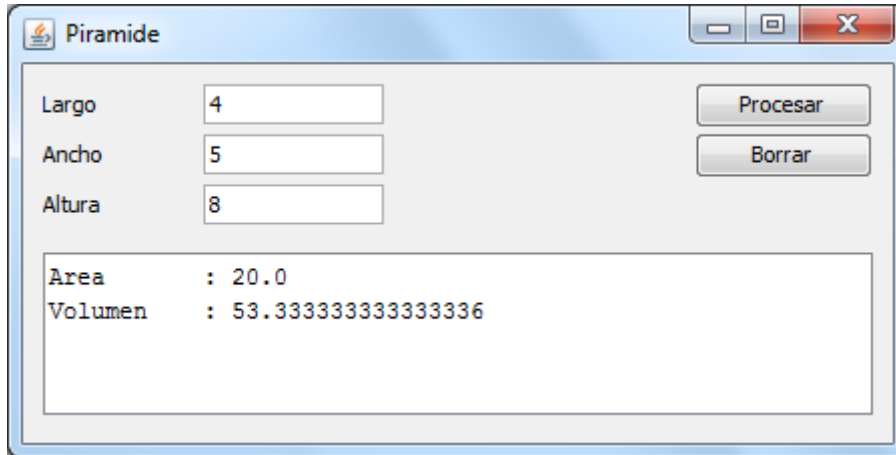
import java.awt.EventQueue;
import java.awt.Font;
import javax.swing.JFrame;

```

```

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

```

```

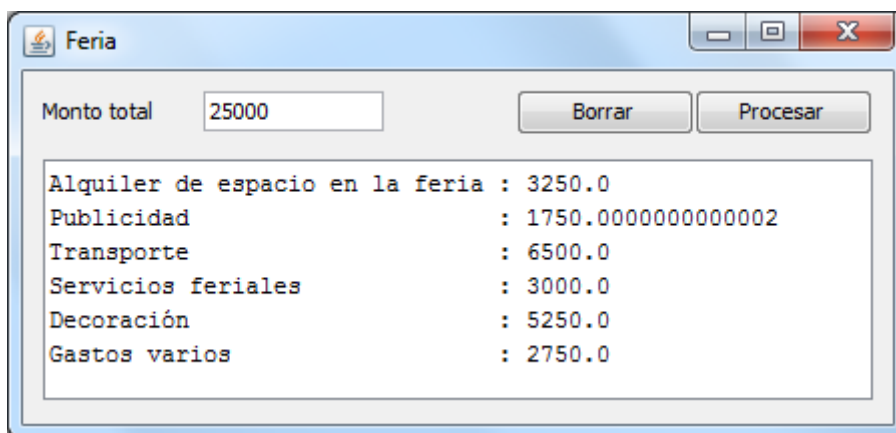
// Entrada de datos
Leer montoTotal

// 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

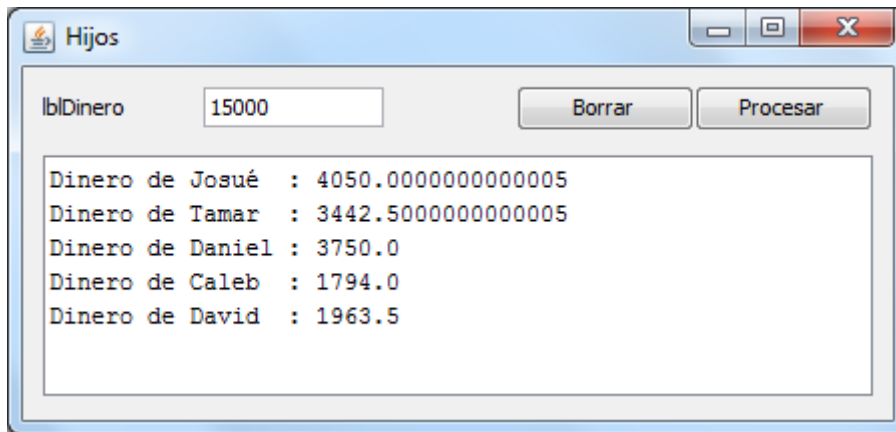
```
// Declaración de variables
real dinero, dinTamar, dinJosue, dinCaleb, dinDaniel, dinDavid

// 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

```

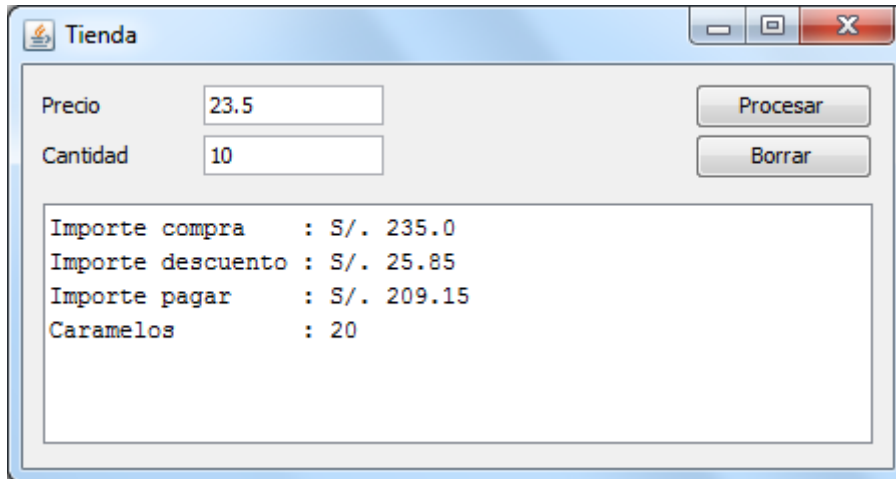
```

// 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

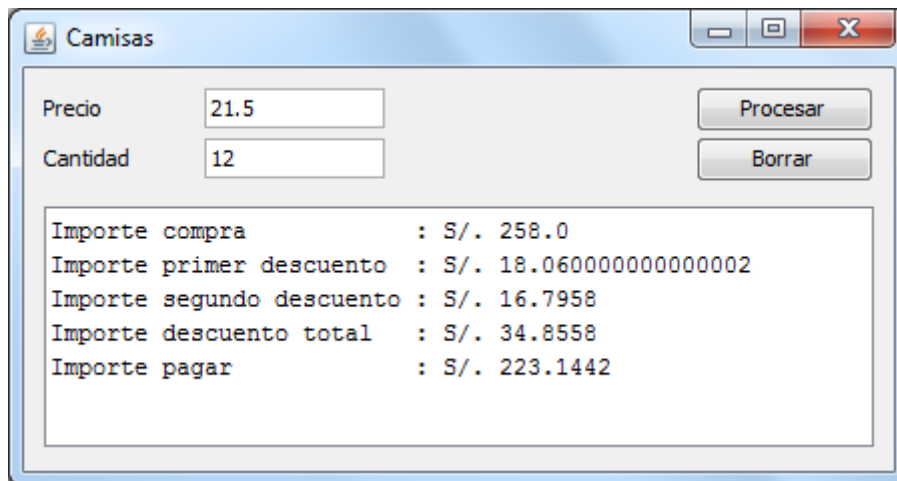
// Entrada de datos
Leer pre, can

// 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

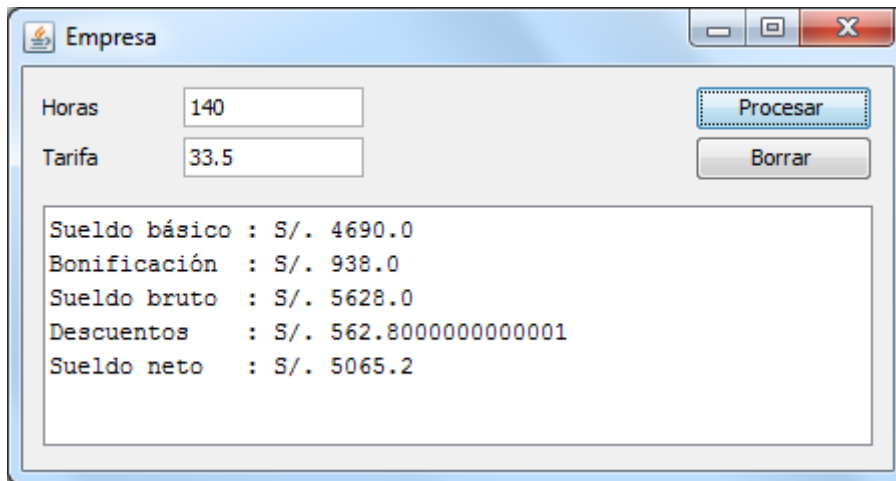
```

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;

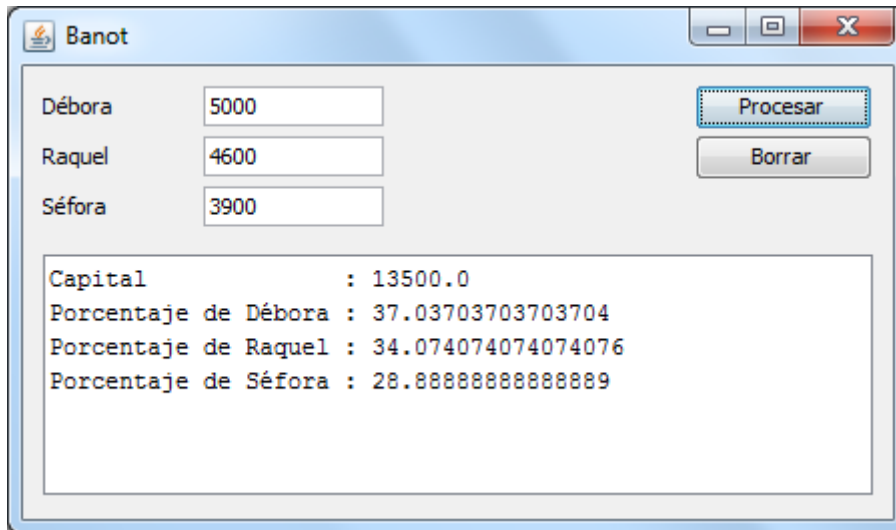
import java.awt.EventQueue;
import java.awt.Font;
import javax.swing.JFrame;
import javax.swing.JButton;

```

```

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

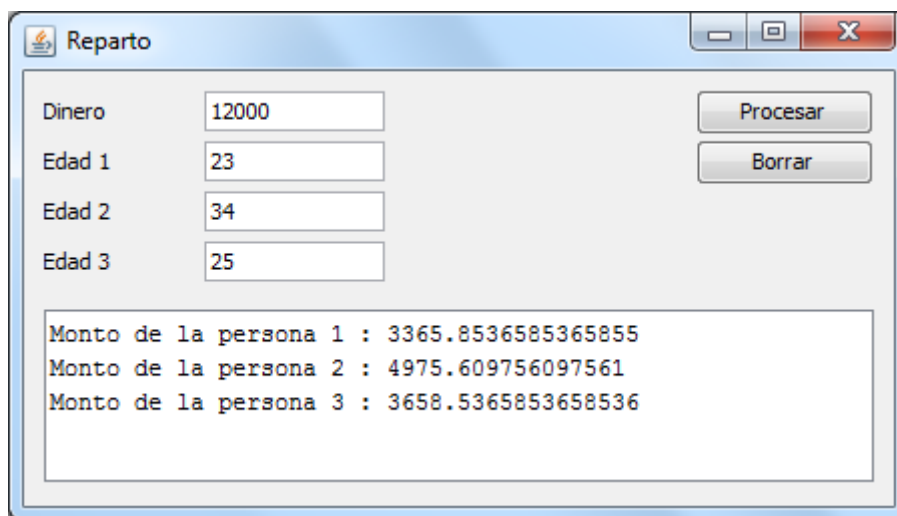
// Calcula la suma total de edades
sumaEdades = edadP1 + edadP2 + edadP3

// 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();
}
}
```