

Introducción a la Algoritmia

Índice

Presentación	5
Red de contenidos	6
Unidad de Aprendizaje 1	
SEMANAS 1 - 2 : Introducción al Lenguaje Java	7
SEMANAS 3 - 4 : Algoritmos y Estructuras de Secuencia	19
Unidad de Aprendizaje 3	
SEMANA 5 : Estructura de Selección Simple if	39
SEMANA 6 : Estructura de Selección Doble if...else	59
SEMANA 7 : Semana de Exámenes Parciales	
SEMANAS 8 : Estructura de Selección Doble Encadenada if...else...if	81
SEMANA 9 : Estructura de Selección Múltiple switch	109
Unidad de Aprendizaje 4	
SEMANAS 10 : Métodos tipo void	125
SEMANAS 11 : Métodos con valor de retorno	151
Unidad de Aprendizaje 5	
SEMANAS 12 : Contadores y acumuladores	173
SEMANAS 13-16 : Estructuras de Repetición	197
SEMANA 17 : Semana de Exámenes Finales	
ANEXO 1 : Paneles	233
ANEXO 2 : Clases Básicas de Swing	239

Presentación

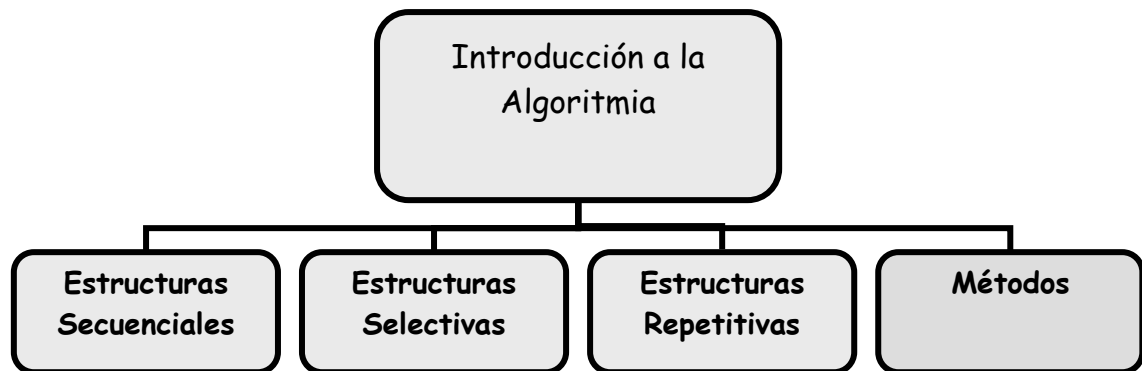
Un algoritmo es una secuencia ordenada y finita de pasos que permite resolver un problema. Puede decirse también que un algoritmo es un método para resolver un problema. El término algoritmia proviene del nombre del gran matemático árabe Al-Khorezmi, el cual escribió sobre los años 800 y 825 su obra *Quitad Al Mugabala*, donde se recogía el sistema de numeración hindú y el concepto del cero.

La algoritmia es un pilar fundamental de las ciencias de la computación puesto que provee métodos de solución de problemas, que serán implementados en los programas de computadora. En este sentido, un programa de computadora es la implementación de un algoritmo en un determinado lenguaje de programación. Este curso es una introducción a la algoritmia y a la programación en Java.

Este manual consta de ocho temas, los cuales serán desarrollados en 16 semanas. Se ha contemplado para ello objetivos concretos y un conjunto de actividades que serán desarrolladas en clase bajo la guía del profesor.

Finalmente, se espera que el alumno valore el material que tiene en sus manos y pueda probar los programas en la máquina.

Red de contenidos



UNIDAD DE
APRENDIZAJE

1

SEMANA

1-2

INTRODUCCIÓN AL LENGUAJE JAVA

LOGRO DE LA UNIDAD DE APRENDIZAJE

Al finalizar la unidad, los alumnos, usando los tipos de datos adecuados y los métodos adecuados del lenguaje, declararán y asignarán variables y, transformarán expresiones aritméticas al lenguaje Java.

TEMARIO

1. El lenguaje Java.
2. Etapas de desarrollo de un programa Java.
3. Tipos de programas Java
4. Software para el desarrollo de programas Java.
5. Identificadores.
6. Palabras reservadas.
7. Tipos de datos primitivos.
8. Variables primitivas.
9. Literales.
10. Sentencias de asignación.
11. Expresiones aritméticas.

ACTIVIDADES

Los alumnos responden un cuestionario de preguntas.

1. EL LENGUAJE JAVA

Java es un lenguaje de programación de alto nivel con el que se pueden escribir programas de cualquier tipo. Se dice que es un lenguaje de alto nivel porque se asemeja más al lenguaje humano, a diferencia de los lenguajes de bajo nivel que se asemejan más al lenguaje de las computadoras (conocidos como lenguajes máquina). Los programas escritos en Java no son comprensibles en forma directa por el procesador, por lo que requieren de pasos intermedios de traducción al lenguaje del computador conocido como lenguaje máquina.

2. ETAPAS DE DESARROLLO DE UN PROGRAMA JAVA

Para desarrollar un programa en Java sigue las siguientes etapas:

2.1 Edición

En esta etapa, se escriben las instrucciones del programa usando el lenguaje Java y se guarda en un archivo cuyo nombre debe terminar con la extensión **.java**. Así, por ejemplo, el archivo podría llamarse **Saludo.java**. A este programa escrito en Java se denomina **código fuente** y para escribirlo se puede recurrir a cualquier editor de texto.

2.2 Compilación

En esta etapa, se compila el código fuente usando el compilador de Java, que es un programa denominado **javac.exe**, con lo que se obtiene un nuevo código conocido como **código de bytes**, que queda guardado en un archivo con el mismo nombre que el archivo de código fuente, pero con la extensión **.class**. Así, si el archivo de código fuente se denomina **Saludo.java**, el *código de bytes* quedará almacenado automáticamente en un archivo denominado **Saludo.class**.

2.3 Ejecución

En esta etapa, el código de bytes es ejecutado por la Máquina Virtual de Java (JVM, siglas en inglés de Java Virtual Machine). El código de bytes es el lenguaje de la JVM. Existe una JVM para cada plataforma (para cada sistema operativo); pero, todas las JVM pueden ejecutar el mismo código de bytes. Así, el código de bytes es independiente de la plataforma. Esto hace que los programas Java puedan ser ejecutados en cualquier máquina que disponga de una JVM. Actualmente, existe una JVM para la mayor parte de las plataformas, lo que garantiza que los programas Java sean ampliamente portables.

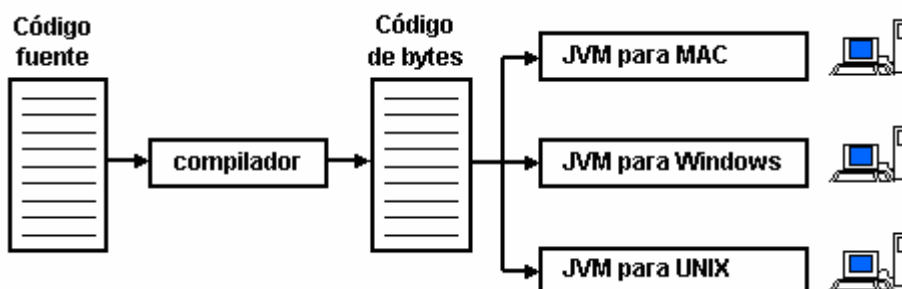


Figura 1.1 Compilación y ejecución de un programa Java

3. TIPOS DE PROGRAMAS JAVA

En Java se pueden crear básicamente dos tipos de programas: *applets* y *aplicaciones*.

Los **applets** son programas Java que se transmiten por Internet y que se ejecutan incrustados en una página Web. Para ejecutar un applet se requiere tener instalada una versión de la JVM. En la Figura 1.2 se muestra un applet en ejecución. Note que un applet no tiene ventana.

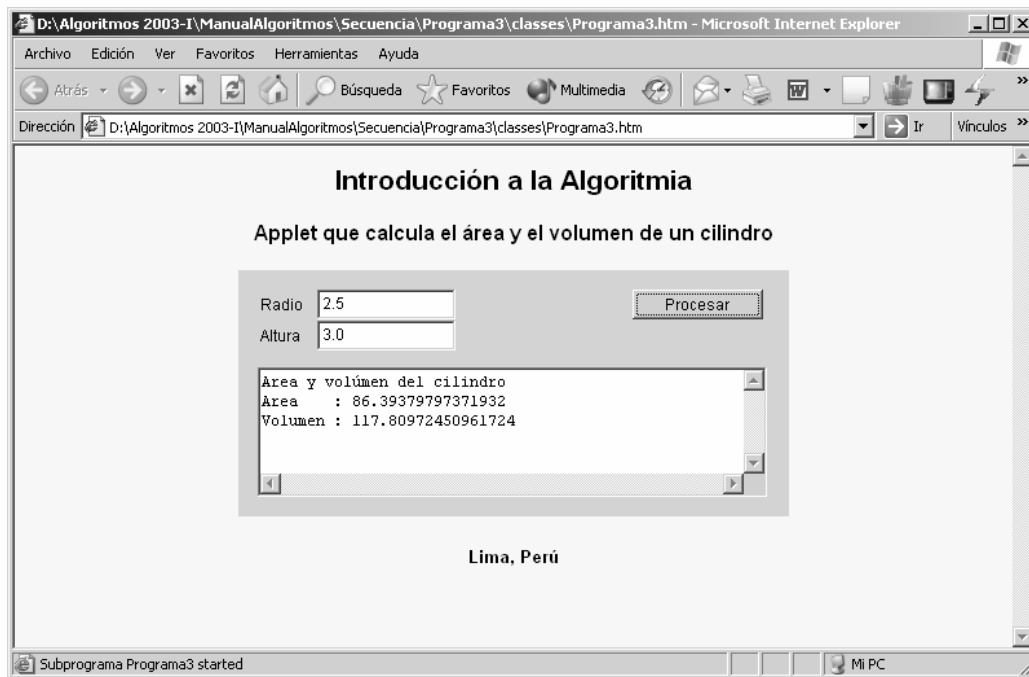


Figura 1.2 Ejemplo de applet ejecutándose dentro de Internet Explorer

Las **aplicaciones** son programas independientes de un navegador Web; pero que, para ser ejecutados, también necesitan de una versión de la JVM. En la Figura 1.3 se muestra una aplicación en ejecución. Note que la aplicación tiene su propia ventana.

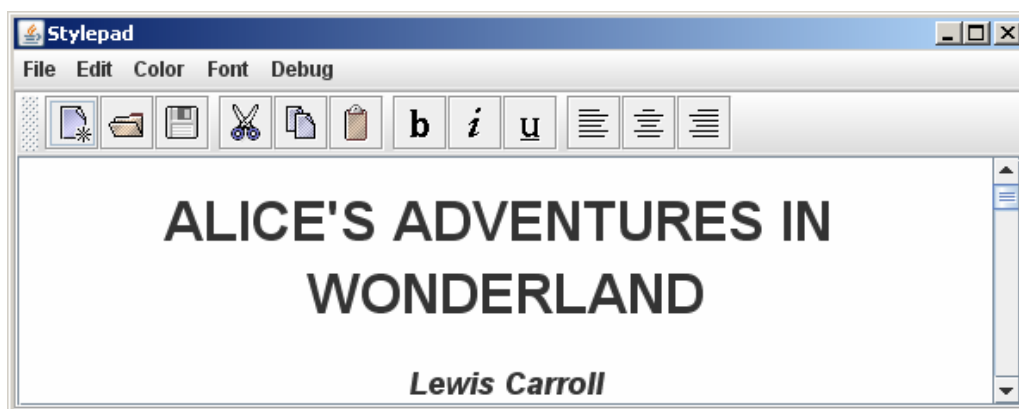


Figura 1.3 Ejemplo de aplicación ejecutándose sin necesidad de un navegador de Internet.

4. SOFTWARE PARA EL DESARROLLO DE PROGRAMAS JAVA

Para el desarrollo de programas Java usaremos el siguiente software:

- **Java SE Development Kit (JDK)**, que contiene el compilador, la JVM y a las librerías del lenguaje. El JDK puede ser descargado de la página de Sun Microsystems cuya dirección es <http://java.sun.com>
- **JCreator LE**, que es un Entorno Integrado de Desarrollo (IDE, siglas del inglés *Integrated Development Environment*) que permite editar, compilar y ejecutar programas Java. Para la compilación y ejecución se requiere tener instalado el JDK. JCreator LE puede ser descargado de la página de Xinox Software cuya dirección es <http://www.jcreator.com>

A parte de JCreator LE existen otros Entornos Integrados de Desarrollo profesionales como es el caso de NetBeans de Sun Microsystems, JBuilder de Borland, Visual Cafe de Symantec, Eclipse de IBM, JDeveloper de Oracle, Visual Age de IBM, etc.

5. IDENTIFICADORES

Los identificadores son *nombres* usados para identificar diversos elementos de programa como variables, constantes, métodos, clases, etc.

Para crear un identificador 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

Ejemplo 1.1

Son válidos los siguientes identificadores:

edadMaxima
edadmaxima
monto_total
sueldo_bruto
suelbru
\$ganancia
nota2
importeCompra

Note que los identificadores **edadMaxima** y **edadmaxima** no son iguales dado que **M** (mayúscula) no es lo mismo que **m** (minúscula).

Ejemplo 1.2

Son inválidos los siguientes identificadores:

1cuenta	No puede comenzar con un número
monto total	Contiene un carácter inválido intermedio (el espacio en blanco)
premio#	Contiene el carácter inválido #
final	No puede ser una palabra reservada
continue	No puede ser una palabra reservada

6. PALABRAS RESERVADAS

Se denominan palabras reservadas a aquellas palabras que tienen un significado especial dentro del lenguaje y que por lo tanto no pueden ser utilizados para otros propósitos. Así, una palabra reservada no se puede utilizar como un identificador.

En la tabla que sigue se muestran las 50 palabras reservadas del lenguaje Java. Note que todas ellas están en minúsculas.

Tabla 1.1 Palabras reservadas del lenguaje Java

abstract	default	if	private	this
boolean	do	implements	protected	throw
break	double	import	public	throws
byte	else	instanceof	return	transient
case	extends	int	short	try
catch	final	interface	static	void
char	finally	long	strictfp	volatile
class	float	native	super	while
const	for	new	switch	assert
continue	goto	package	synchronized	enum

7. TIPOS DE DATOS PRIMITIVOS

Se conocen como tipos de datos primitivos a los tipos de datos que forman parte del núcleo del lenguaje Java. Existen ocho tipos primitivos que se muestran en la tabla 1.2

Tabla 1.2 Tipos de datos primitivos de java

Dato	Tipo	Bits	Rango
carácter	char	16	0 a 65535
entero	byte	8	-128 a 127
	short	16	-32768 a 32767
	int	32	-2147483648 a 2147483647
	long	64	-9223372036854775808 a 9223372036854775807
real	float	32	-3.4×10^{38} a -1.4×10^{-45} , 1.4×10^{-45} a 3.4×10^{38}
	double	64	-1.7×10^{308} a -4.9×10^{-324} , 4.9×10^{-324} a 1.7×10^{308}
booleano	boolean	1	true, false

A parte de los tipos primitivos existen otros tipos conocidos como tipos referencia a los que se conoce también como tipos abstractos o simplemente clases. Para el manejo de cadenas de caracteres el lenguaje Java dispone del tipo referencia **String**.

8. VARIABLES PRIMITIVAS

Una variable es una localización de memoria en la que se puede almacenar un valor

que puede cambiar en el transcurso de la ejecución de un programa. Una variable es primitiva si el valor que almacena es de tipo primitivo.

Todas las variables deben ser declaradas antes de ser utilizadas. Para declarar una variable se usa una sentencia de declaración que en su forma básica tiene el siguiente formato:

tipo nombre;

Esta forma básica de declaración puede repetirse para cada variable a declarar; sin embargo, si varias variables comparten el mismo tipo de dato, puede usarse el siguiente formato:

tipo nombre1, nombre2, ...,nombren;
--

Ejemplo 1.3

```
int edad;
int hijos;
double sueldo;
double bonificacion;
char letra;
String curso;
```

Ejemplo 1.4

```
int edad, hijos;
double sueldo, bonificacion;
int a, b, c;
```

9. LITERALES

Una literal es la representación de un valor en el código fuente del programa.

9.1 Literales enteros

Cualquier valor numérico entero es un literal entero. Los literales enteros se consideran de tipo **int**. Para especificar que un literal es de tipo **long**, debe añadirse, como sufijo, la letra **L** ó **l**.

Por ejemplo

- Los siguientes literales son de tipo **int**

12, 34, 0, -50, etc.

- Los siguientes literales son de tipo **long**

9223372036854775807L, 25L, -1L, etc.

9.2 Literales reales

Cualquier valor numérico decimal con parte fraccionaria es un literal real. Los literales reales se consideran de tipo **double**. Para especificar que un literal es de tipo **float**, debe añadirse, como sufijo, la letra **F** ó **f**.

Por ejemplo:

- Los siguientes literales son de tipo **double**

`1.23, 3.456, -2.0, 3.25E+12, 2.7e-5, etc.`

- Los siguientes literales son de tipo **float**

`2.75f, -4.567f, 2.0F, 6.73e+2f, etc.`

Para representar un literal real en notación científica se usa la letra **E** ó **e** para expresar la potencia de 10.

Por ejemplo

`3.25E+12` representa a 3.25×10^{12}

`2.7E-5` representa a 2.7×10^{-5}

El signo + que acompaña al exponente es opcional.

9.3 Literales booleanos

Los únicos literales booleanos son los siguientes:

`true, false`

9.4 Literales de carácter

Un literal de carácter consiste de un único carácter encerrado dentro de un par de comillas simples.

`'a', '1', '2', '$', etc.`

Una secuencia de escape es un conjunto de caracteres

9.5 Literales de Cadena

Un literal de cadena consiste de un conjunto de caracteres encerrados entre comillas dobles.

`"Hola mundo"
"Bienvenido a Java"
"Algoritmos Computacionales"
"abcde123xy"
"Edad inválida"
etc`

10. SENTENCIAS DE ASIGNACION

Una sentencia de asignación se utiliza para asignar (almacenar) un valor a una variable. En una sentencia de asignación el valor situado a la derecha del signo igual se almacena en la variable situada a la izquierda del signo igual.

Una sentencia de asignación tiene la siguiente forma:

variable = expresión;

Donde expresión puede ser una variable, un literal o una combinación de variables, literales y operadores. La sentencia de asignación almacena en variable el valor de la expresión; para esto se requiere que expresión y variable tengan el mismo tipo de dato.

Notas

- La asignación tiene *carácter destructivo*. Esto es, la variable que recibe la asignación pierde su valor actual de forma irrecuperable
- Una variable de tipo double puede recibir la asignación de un valor de tipo int. Esta es una excepción a la regla. En ese caso el valor de tipo int se convierte automáticamente al tipo double antes de ser asignado.
- Es un error de sintáxis tratar de asignar un valor de tipo double a una variable de tipo int.

Ejemplo 1.5

```
// Declara las variables p, q y r
int p, q, r;
```

```
// Asigna el valor 2 a la variable p
p = 2;
```

```
// Asigna una copia del valor de p a la variable q
q = p;
```

```
// Evalúa el valor de la expresión 2*p + q a la variable r
r = 2*p + q;
```

p	<div style="border: 1px solid black; padding: 2px 10px;">2</div>
q	<div style="border: 1px solid black; padding: 2px 10px;">2</div>
r	<div style="border: 1px solid black; padding: 2px 10px;">6</div>

11. EXPRESIONES ARITMETICAS

Una expresión aritmética es una combinación de variables, literales y operadores aritméticos.

11.1 Operadores Aritméticos

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

Tabla 1.2 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 1.6

$2 + 5$	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

11.2 Reglas de jerarquía de los operadores aritméticos

Cuando una expresión aritmética tiene más de un operador aritmético, el orden de aplicación de los operadores sigue un orden preciso determinado por las *reglas jerarquía de los operadores aritméticos* que se muestran en la siguiente tabla:

Tabla 1.3 Jerarquía de los operadores aritméticos

Operador	Precedencia
()	Se evalúan en primer lugar.
* / %	Se evalúan en segundo lugar.
+ -	Se evalúan al último.

Si existen paréntesis anidados, se evalúa primero la expresión en el par más interno. Si varios operadores o paréntesis tienen la misma precedencia, es decir, están en el mismo nivel de jerarquía, la evaluación será de izquierda a derecha.

Ejemplo 1.7

Escriba en Java las siguientes expresiones algebraicas:

$$a. \quad e = a^2 + 3bc + 2$$

$$b. \quad z = \frac{a+b+2}{a^2+1} + 2ab$$

$$c. \quad prom = \frac{n1+n2+n3+n4}{4}$$

Solución

$$a. \quad e = a*a + 3*b*c + 2;$$

$$b. \quad z = (a+b+2)/(a*a+1) + 2*a*b;$$

$$c. \quad prom = (n1+n2+n3+n4)/4;$$

11.3 Métodos matemáticos

En la siguiente tabla se listan algunos métodos matemáticos del lenguaje Java:

Álgebra	Método Java
n^m	Math.pow(n, m)
\sqrt{n}	Math.sqrt(n)

Donde n y m pueden ser de tipo **int** o de tipo **double**; pero el resultado en ambos casos es de tipo **double**.

Ejemplo 1.8

Escriba en Java las siguientes expresiones algebraicas:

$$a. \quad e = a^7 + (a+b^2)^4 + \frac{\sqrt{2+b}}{a^2+b^2}$$

$$b. \quad e = \frac{(a+b^7)^5}{\sqrt[3]{a+b^2}}$$

Solución:

$$a. \quad e = \text{Math.pow}(a,7) + \text{Math.pow}(a + b * b,4) + \text{Math.sqrt}(2 + b)/(a * a + b * b);$$

$$b. \quad e = \text{Math.pow}(a + \text{Math.pow}(b,7),5)/\text{Math.pow}(a + b * b,1/3.0);$$

Problemas propuestos

1. Ponga un aspa al costado de los identificadores válidos.

- 1.1. año nacimiento
- 1.2. sueldoNeto
- 1.3. \$monto
- 1.4. peso-máximo
- 1.5. EDAD_PROMEDIO
- 1.6. númeroDeEmpleados
- 1.7. _temperatura
- 1.8. xy1\$\$\$\$\$
- 1.9. 1_abc\$\$
- 1.10. volumen_CILINDRO

2. Declare variables adecuadas para almacenar los siguientes datos:

2.1. El área de un círculo.

2.2. El sexo de una persona.

2.3. La estación del año.

2.4. El importe bruto, el importe de descuento y el importe neto de una compra.

3. Escriba las siguientes expresiones algebraicas en el lenguaje Java:

3.1.
$$z = \frac{ab}{\frac{c}{d} + ab^2 + \frac{a+b}{c+d}} + x$$

3.2.
$$e = \sqrt{a+2} + \frac{\sqrt[3]{a+b+1}}{ab}$$

3.3.
$$e = \frac{abc}{a+b} + \frac{a^2 - b}{\frac{c}{e}}$$

$$3.4. \quad x = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

$$3.5. \quad e = \frac{a+b}{cd^5} + a^8b$$

Para recordar

- Una variable primitiva almacena un valor primitivo.
- Un literal es la representación de un valor en el código fuente de un programa.
- La asignación tiene carácter destructivo.
- 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.
- La evaluación de expresiones aritméticas sigue un orden dado por las reglas de jerarquía de los operadores aritméticos.

UNIDAD DE
APRENDIZAJE

2

SEMANA

3-4

ESTRUCTURAS DE SECUENCIA

LOGRO DE LA UNIDAD DE APRENDIZAJE

Al finalizar la unidad, los alumnos, empleando las etapas de desarrollo de un algoritmo y las instrucciones algorítmicas básicas, diseñarán algoritmos en pseudocódigo que resuelvan problemas que involucren instrucciones secuenciales.

TEMARIO

1. Introducción.
2. Algoritmo.
3. Pseudocódigo.
4. Instrucciones algorítmicas básicas.
5. Estructuras de secuencia.

ACTIVIDADES

Los alumnos desarrollan algoritmos que involucren estructuras de secuencia.

1. ALGORITMO

Un algoritmo es un método de solución de un problema expresado a través de un conjunto de pasos, procedimientos o acciones. Por ejemplo, es un algoritmo el conjunto de instrucciones que deben seguirse para instalar una impresora. También es un algoritmo una receta para preparar un plato de comida. Otro caso de algoritmo es el algoritmo de Euclides para la obtención del máximo común divisor de dos números. Un programa, a su vez, es la implementación de un algoritmo mediante un lenguaje de programación de forma que sea entendible por el computador. En la figura 2.1 puede verse la relación entre problema, algoritmo y programa.

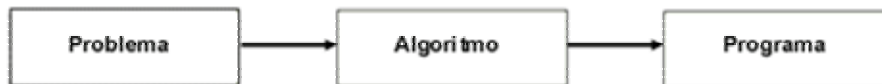


Figura 2.1 Problema, Algoritmo y Programa

2. PSEUDOCODIGO

El pseudocódigo es un lenguaje algorítmico informal que es una imitación de uno o más lenguajes de programación de alto nivel. 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).

Al contrario que los lenguajes de programación de alto nivel como Java, C, C++, Pascal, etc., no existe un conjunto de reglas que definan con precisión lo que es y lo que no es un pseudocódigo. En otras palabras, no existe un pseudocódigo estándar. Varía de un programador a otro.

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.

Ejemplo 2.1

Diseñe un algoritmo que determine el monto a pagar por la compra de cierta cantidad de unidades de un producto.

Solución:

```

Inicio
  // Declaración de variables
  entero cantidad
  real precio, montopagar

  // Entrada de datos
  Leer precio, cantidad

  // Proceso de cálculo
  montopagar = precio*cantidad

  // Salida de resultados
  Imprimir montopagar
Fin
  
```

3. ETAPAS DE DESARROLLO DE UN ALGORITMO

A los algoritmos que pueden ser ejecutados por un computador se denominan algoritmos computacionales. Todo algoritmo computacional puede dividirse en cuatro etapas:

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

4. INSTRUCCIONES ALGORITMICAS BASICAS

Existen cuatro instrucciones algorítmicas básicas que son las siguientes:

- **Declaración de variables:-** En esta etapa se declaran las variables que serán utilizadas en el algoritmo. Para esto, imitemos la forma de declarar variables del lenguaje Java usando los *tipos de datos algorítmicos* mostrados en la siguiente tabla:

Tabla 2.1 Tipos de datos algorítmicos

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

- **Entrada:-** La entrada consiste en obtener un dato desde algún dispositivo de entrada y trasladarlo a la memoria para ser almacenada en una variable. 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.

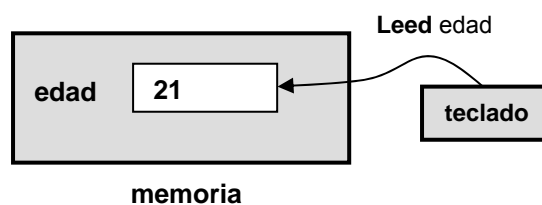


Figura 2.2 Entrada

- **Salida:-** La salida consiste en trasladar a algún dispositivo de salida el valor de una variable. 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.

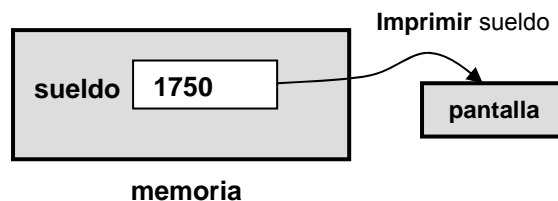


Figura 2.3 Salida

- **Asignación:-** La asignación consiste en dar un valor a una variable. La forma general de asignación es la siguiente:

variable = expresión

Donde expresión puede ser una variable, un literal o una combinación de variables, literales y operadores.

5. ESTRUCTURAS DE SECUENCIA

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

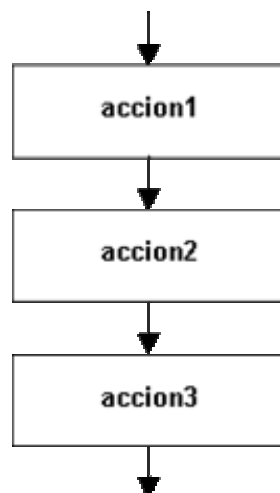


Figura 2.4 Estructura de Secuencia

6. PROBLEMAS RESUELTOS

Problema 2.1

El cálculo del pago mensual de un empleado de una empresa se efectúa de la siguiente manera: el sueldo básico se calcula en base al número total de horas

trabajadas basado en una tarifa horaria; al sueldo básico, se le aplica una bonificación del 20% obteniéndose el sueldo bruto; al sueldo bruto, se le aplica un descuento del 10% obteniéndose el sueldo neto. Escriba un programa que calcule e imprima el sueldo básico, el sueldo bruto y el sueldo neto de un trabajador.

Algoritmo

Inicio

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

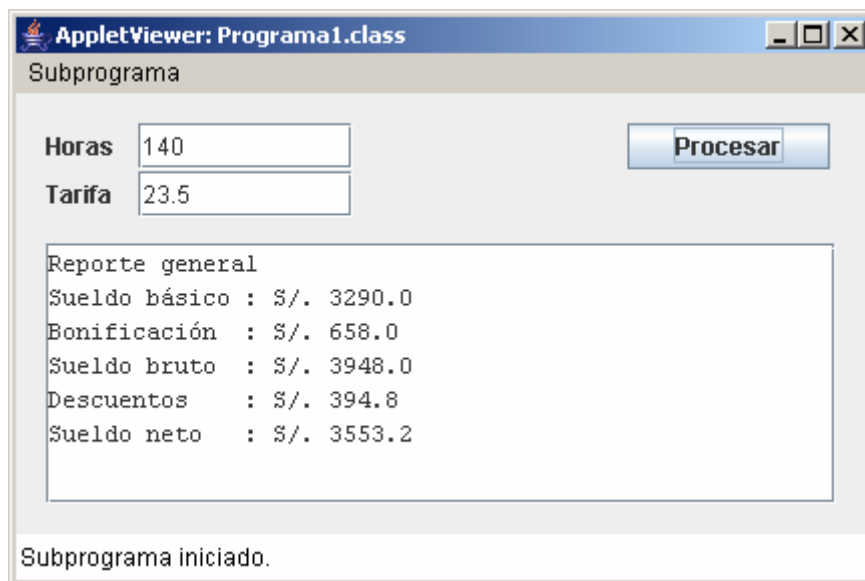
// Entrada de datos
Leer horasTrab, tarifaHor

// 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



```
import java.awt.event.*;
import java.awt.*;
import javax.swing.*;

public class Programa1 extends JApplet implements ActionListener{
    // Declaración de variables
    JButton btnProcesar;
    JTextArea txtS;
    JLabel lblHoras;
    JLabel lblTarifa;
    JTextField txtHoras;
    JTextField txtTarifa;
    JScrollPane scpScroll;
```

```

// Crea la interfaz gráfica de usuario GUI
public void init() {
    getContentPane().setLayout(null);

    lblHoras = new JLabel("Horas");
    lblHoras.setBounds(15, 15, 42, 23);
    getContentPane().add(lblHoras);

    lblTarifa = new JLabel("Tarifa");
    lblTarifa.setBounds(15, 39, 42, 23);
    getContentPane().add(lblTarifa);

    txtHoras = new JTextField();
    txtHoras.setBounds(61, 15, 107, 23);
    getContentPane().add(txtHoras);

    txtTarifa = new JTextField();
    txtTarifa.setBounds(61, 39, 107, 23);
    getContentPane().add(txtTarifa);

    btnProcesar = new JButton("Procesar");
    btnProcesar.setBounds(305, 15, 101, 23);
    btnProcesar.addActionListener(this);
    getContentPane().add(btnProcesar);

    txtS = new JTextArea();
    txtS.setFont(new Font("monospaced", 0, 12));

    scpScroll = new JScrollPane(txtS);
    scpScroll.setBounds(15, 75, 394, 130);
    getContentPane().add(scpScroll);
}

// Procesa eventos de tipo ActionEvent
public void actionPerformed( ActionEvent e ){

    // Declaración de variables
    double horasTrab, tarifaHor;
    double sueldoBas, montoBoní, sueldoBru, montoDesc, sueldoNet;

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

    // Cálculo de montos
    sueldoBas = horasTrab*tarifaHor;
    montoBoní = 0.20*sueldoBas;
    sueldoBru = sueldoBas+montoBoní;
    montoDesc = 0.10*sueldoBru;
    sueldoNet = sueldoBru-montoDesc;

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

```

Problema 2.2

Diseñe un programa que permita convertir una cantidad dada en metros a sus equivalentes en centímetros, pulgadas, pies y yardas. Considere la siguiente información:

1 metro = 100 centímetros
 1 pie = 12 pulgadas
 1 yarda = 3 pies
 1 pulgada = 2.54 centímetros

Algoritmo

Inicio

```

// Declaración de variables
real xmetros, xpies, xpulgadas, xyardas, xcentimetros

// Entrada de datos
Leer xmetros

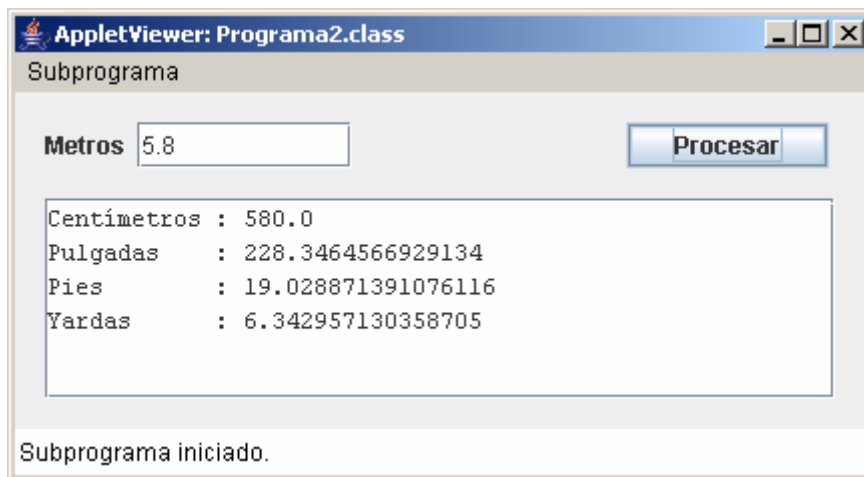
// Proceso de conversión
xcentimetros = xmetros*100
xpulgadas = xcentimetros/2.54
xpies = xpulgadas/12
xyardas = xpies/3

// Salida de resultados
Imprimir xcentimetros, xpulgadas, xpies, xyardas

```

Fin

Programa:



```

import java.awt.*;
import javax.swing.*;
import java.awt.event.*;

public class Programa2 extends JApplet implements ActionListener{
    // Declaración de variables
    JLabel lblMetros;
    JButton btnProcesar;
    JTextArea txtS;
    JTextField txtMetros;
    JScrollPane scpScroll;

    // Crea la interfaz gráfica de usuario
    public void init() {
        getContentPane().setLayout(null);

        lblMetros= new JLabel("Metros");
        lblMetros.setBounds(15, 15, 42, 23);
        getContentPane().add(lblMetros);

        txtMetros = new JTextField();
    }
}

```

```

        txtMetros.setBounds(61, 15, 107, 23);
        getContentPane().add(txtMetros);

        btnProcesar = new JButton("Procesar");
        btnProcesar.setBounds(305, 15, 101, 23);
        btnProcesar.addActionListener(this);
        getContentPane().add(btnProcesar);

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

        scpScroll = new JScrollPane(txtS);
        scpScroll.setBounds(15, 53, 394, 100);
        getContentPane().add(scpScroll);
    }

    // Procesa eventos de tipo(ActionEvent
    public void actionPerformed( ActionEvent e ){

        // Declaración de variables
        double xmetros, xpies, xpulgadas, xyardas, xcentimetros;

        // Entrada de datos
        xmetros = Double.parseDouble(txtMetros.getText());

        // Proceso de conversión
        xcentimetros = xmetros*100;
        xpulgadas = xcentimetros/2.54;
        xpies = xpulgadas/12;
        xyardas = xpies/3;

        // Salida de resultados
        txtS.setText( "Centímetros : " + xcentimetros + "\n" );
        txtS.append ( "Pulgadas      : " + xpulgadas + "\n" );
        txtS.append ( "Pies          : " + xpies + "\n" );
        txtS.append ( "Yardas        : " + xyardas);
    }
}

```

Problema 2.3

Escriba un programa que calcule el área total y el volumen de un cilindro. Considere las siguientes fórmulas: $A = 2\pi r(r+h)$ y $V = \pi r^2 h$; siendo A el área, V el volumen, r el radio y h la altura.

Algoritmo

```

Inicio
    // Declaración de variables
    real r, h, area, volumen

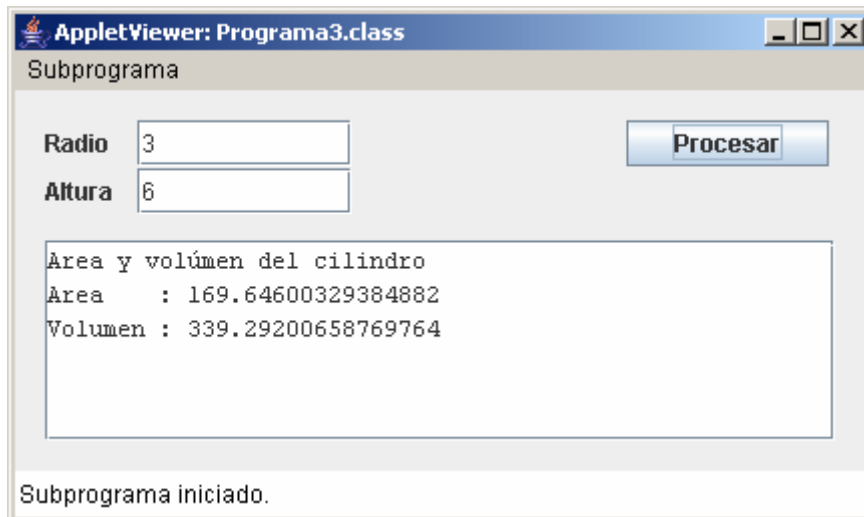
    // Entrada de datos
    Leer r, h

    // Proceso de cálculo
    area      = 2*3.1416*r*(r+h)
    volumen   = 2*3.1416*r*r*h

    // Salida de resultados
    Imprimir area, volumen
Fin

```

Programa



```
import java.awt.event.*;
import java.awt.*;
import javax.swing.*;

public class Programa3 extends JApplet implements ActionListener{
    // Declaración de variables
    JButton btnProcesar;
    JTextArea txtS;
    JLabel lblAltura;
    JLabel lblRadio;
    JTextField txtAltura;
    JTextField txtRadio;
    JScrollPane scpScroll;

    // Crea la interfaz gráfica de usuario GUI
    public void init() {
        getContentPane().setLayout(null);

        lblAltura = new JLabel("Altura");
        lblAltura.setBounds(15, 39, 42, 23);
        getContentPane().add(lblAltura);

        lblRadio = new JLabel("Radio");
        lblRadio.setBounds(15, 15, 42, 23);
        getContentPane().add(lblRadio);

        txtRadio = new JTextField();
        txtRadio.setBounds(61, 15, 107, 23);
        getContentPane().add(txtRadio);

        txtAltura = new JTextField();
        txtAltura.setBounds(61, 39, 107, 23);
        getContentPane().add(txtAltura);

        btnProcesar = new JButton("Procesar");
        btnProcesar.setBounds(305, 15, 101, 23);
        btnProcesar.addActionListener(this);
        getContentPane().add(btnProcesar);

        txtS = new JTextArea();
        txtS.setFont(new Font("monospaced", 0, 12));

        scpScroll = new JScrollPane(txtS);
        scpScroll.setBounds(15, 75, 394, 100);
        getContentPane().add(scpScroll);
    }

    // Procesa eventos de tipo(ActionEvent)
```

```

public void actionPerformed( ActionEvent e ){

    // Declaración de variables
    double r, h, area, volumen;

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

    // Proceso
    area = 2*Math.PI*r*(r+h);
    volumen = 2*Math.PI*r*r*h;

    // Salida de resultados
    txtS.setText("Area : " + area + "\n");
    txtS.append ("Volumen : " + volumen);

}
}

```

Problema 2.4

Débora, Raquel y Séfora aportan cantidades de dinero para formar un capital. Diseñe un programa que determine el capital formado y el porcentaje de dicho capital que aporta cada uno.

Algoritmo

```

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

    // Entrada de datos
    Leer dineDeb, dineRaq, dineSef

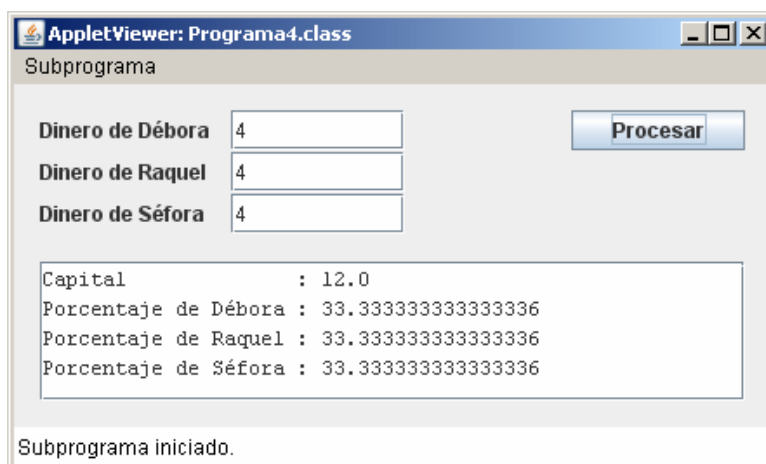
    // Halla el capital formado
    capital = dineDeb + dineRaq + dineSef

    // Determine los porcentajes de cada capital respecto del total
    porcDeb = dineDeb*100/capital
    porcRaq = dineRaq*100/capital
    porcSef = dineSef*100/capital

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

```

Programa



AppletViewer: Programa4.class

Subprograma

Dinero de Débora: 4

Dinero de Raquel: 4

Dinero de Séfora: 4

Procesar

Capital : 12.0
 Porcentaje de Débora : 33.33333333333336
 Porcentaje de Raquel : 33.33333333333336
 Porcentaje de Séfora : 33.33333333333336

Subprograma iniciado.

```
import java.awt.event.*;
import java.awt.event.*;
import java.awt.*;
import javax.swing.*;

public class Programa4 extends JApplet implements ActionListener{
    // Declaración de variables globales
    JButton btnProcesar;
    JTextArea txtS;
    JLabel lblDineroDebora, lblDineroRaquel, lblDineroSefora;
    JTextField txtDineroDebora, txtDineroRaquel, txtDineroSefora;
    JScrollPane scpScroll;

    // Crea la interfaz gráfica de usuario
    public void init() {
        getContentPane().setLayout(null);

        txtDineroDebora = new JTextField();
        txtDineroDebora.setBounds(125, 15, 100, 23);
        getContentPane().add(txtDineroDebora);

        txtDineroRaquel = new JTextField();
        txtDineroRaquel.setBounds(125, 39, 100, 23);
        getContentPane().add(txtDineroRaquel);

        txtDineroSefora = new JTextField();
        txtDineroSefora.setBounds(125, 63, 100, 23);
        getContentPane().add(txtDineroSefora);

        btnProcesar = new JButton("Procesar");
        btnProcesar.setBounds(321, 15, 100, 23);
        btnProcesar.addActionListener(this);
        getContentPane().add(btnProcesar);

        lblDineroDebora = new JLabel("Dinero de Débora");
        lblDineroDebora.setBounds(15, 15, 110, 23);
        getContentPane().add(lblDineroDebora);

        lblDineroRaquel = new JLabel("Dinero de Raquel");
        lblDineroRaquel.setBounds(15, 39, 110, 23);
        getContentPane().add(lblDineroRaquel);

        lblDineroSefora = new JLabel("Dinero de Séfora");
        lblDineroSefora.setBounds(15, 63, 110, 23);
        getContentPane().add(lblDineroSefora);

        txtS = new JTextArea();
        txtS.setFont(new Font("monospaced", 0, 12));

        scpScroll = new JScrollPane(txtS);
        scpScroll.setBounds(15, 102, 406, 150);
        getContentPane().add(scpScroll);
    }

    // Procesa eventos de tipo ActionEvent
    public void actionPerformed( ActionEvent e ){

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

        // Entrada de datos
        dineDeb = Double.parseDouble(txtDineroDebora.getText());
        dineRaq = Double.parseDouble(txtDineroRaquel.getText());
        dineSef = Double.parseDouble(txtDineroSefora.getText());

        // Determina el cantidad 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);
    }
}

```

Problema 2.5

Diseñe un algoritmo que lea un número entero de cinco cifras y determine la cifra central del número. Así, si el número ingresado fuera 45781, la cifra central a mostrar es 7.

Algoritmo

```

Inicio
    // Declaración de variables
    entero numero, centro

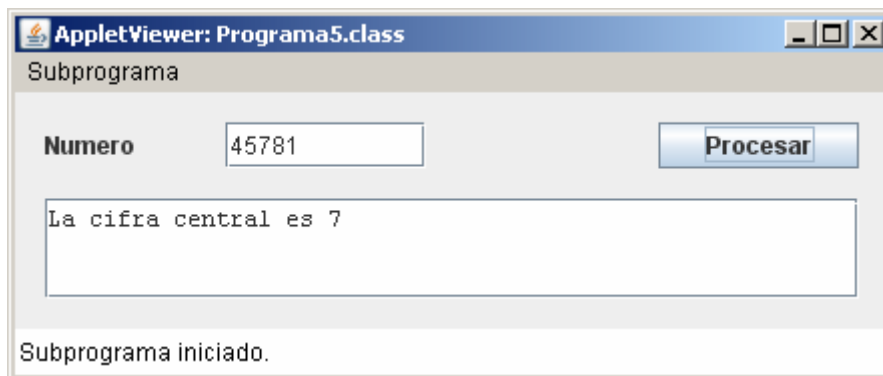
    // Entrada de datos
    Leer numero

    // Determina la cifra central
    centro = (numero%1000)/100

    // Salida de resultados
    Imprimir centro
Fin

```

Programa



```

import java.awt.event.*;
import java.awt.*;
import javax.swing.*;

public class Programa5 extends JApplet implements ActionListener{
    // Declaración de variables
    JButton btnProcesar;
    JTextArea txtS;
    JLabel lblNumero;
    JTextField txtNumero;
    JScrollPane scpScroll;

    // Crea la interfaz gráfica de usuario
    public void init() {

```

```

        getContentPane().setLayout(null);

        txtNumero = new JTextField();
        txtNumero.setBounds(105, 15, 100, 23);
        getContentPane().add(txtNumero);

        btnProcesar = new JButton("Procesar");
        btnProcesar.setBounds(321, 15, 100, 23);
        btnProcesar.addActionListener(this);
        getContentPane().add(btnProcesar);

        lblNumero = new JLabel("Numero");
        lblNumero.setBounds(15, 15, 90, 23);
        getContentPane().add(lblNumero);

        txtS = new JTextArea();
        txtS.setFont(new Font("monospaced", 0, 12));

        scpScroll = new JScrollPane(txtS);
        scpScroll.setBounds(15, 53, 406, 50);
        getContentPane().add(scpScroll);
    }

    // Procesa eventos de tipo ActionEvent
    public void actionPerformed( ActionEvent e ){

        // Declaración de variables
        int numero, centro;

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

        // Determina la cifra central
        centro = (numero%1000)/100;

        // Salida de resultados
        txtS.setText("La cifra central es " + centro);
    }
}

```

Problema 2.6

Diseñe un algoritmo para repartir una cantidad de dinero a tres personas en forma proporcional a sus edades. El monto que le corresponde a cada persona se calcula con la siguiente fórmula:

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

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
```

Problema 2.7

Una tienda ha puesto en oferta la venta de un producto ofreciendo un 11% de descuento sobre el importe de la compra. Diseña un algoritmo 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
real precio, importecom, importedes, importepag
entero unidades

// Entrada de datos
Leer precio, unidades

// Cálculo de importes
importecom = unidades*precio
importedes = 0.11*importecom
importepag = importecom - importedes

// Salida de resultados
Imprimir importecom, importedes, importepag
Fin
```


Problemas propuestos

Actividad

1. Diseñe un algoritmo que determine el área y el perímetro de un rectángulo, sabiendo que:

$$\begin{aligned}\text{area} &= b \times h \\ \text{perimetro} &= 2 \times (b + h)\end{aligned}$$

Siendo b la base y h la altura.

2. Diseñe un algoritmo que determine el área lateral, el área total y el área de la base de un cilindro, sabiendo que:

$$\begin{aligned}\text{areabase} &= \pi \times r^2 \\ \text{arealateral} &= 2 \times \pi \times r \times h \\ \text{areatotal} &= 2 \times \text{areabase} + \text{arealateral}\end{aligned}$$

Siendo r el radio y h la altura.

3. Una institución social tiene un centro de salud, un comedor infantil, una escuela infantil y un asilo de ancianos. La institución recibe anualmente una donación que lo reparte de la siguiente forma: 25% de la donación para la implementación del centro de salud, 35% de la donación para el comedor infantil, 25% de la donación para la escuela infantil y el resto para el asilo de ancianos. Diseñe un algoritmo para efectuar el reparto de la donación.
4. Dada la longitud de un cable en metros, diseñe un algoritmo que exprese dicha longitud en pies y en yardas. Considere los siguientes factores de conversión:

$$\begin{aligned}1 \text{ metro} &= 100 \text{ centímetros} \\ 1 \text{ pulgada} &= 2.54 \text{ centímetros} \\ 1 \text{ yarda} &= 3 \text{ pies} \\ 1 \text{ pie} &= 12 \text{ pulgadas}\end{aligned}$$

5. Se cuenta con tres cantidades de dinero en soles, dólares y marcos, respectivamente. Diseñe un algoritmo que determine el monto total del dinero en euros. Considere los siguientes tipos de cambio:

$$\begin{aligned}1 \text{ dólar} &= 3.51 \text{ soles} \\ 1 \text{ dólar} &= 1.09 \text{ euros} \\ 1 \text{ dólar} &= 2.12 \text{ marcos}\end{aligned}$$

6. Dado un tiempo en segundos, diseñe un algoritmo que exprese dicho tiempo en el formato HH:MM:SS. Por ejemplo, si el tiempo es 14600 segundos, el algoritmo deberá mostrar 4:3:20.
7. Dado un número natural de 4 cifras, diseñe un algoritmo que determine la suma y el producto de las cifras del número.

Autoevaluación

1. Diseñar un programa para convertir grados sexagesimales (S) a grados centesimales (C) y radianes (R). Considere las siguientes fórmulas:

$$S/180 = C/200$$

$$S/180 = R/\pi.$$

2. Escriba un programa 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:

$$F = 9C/5 + 32$$

$$K = R - 187$$

$$R = C + 460$$

3. Diseñe un algoritmo para convertir una cantidad dada en pulgadas a pies, a yardas, a centímetros y a metros. Considere que:

$$1 \text{ yarda} = 3 \text{ pies}$$

$$1 \text{ pie} = 12 \text{ pulgadas}$$

$$1 \text{ pulgada} = 2.54 \text{ centímetros}$$

$$1 \text{ metro} = 100 \text{ centímetros}$$

4. Diseñe un programa que lea la capacidad de un disco duro en gigabytes y lo convierta a megabytes, a kilobytes y a bytes.

$$1 \text{ kilobyte} = 1024 \text{ bytes}$$

$$1 \text{ megabyte} = 1024 \text{ kilobytes}$$

$$1 \text{ gigabyte} = 1024 \text{ megabytes}$$

5. Dada una cantidad de dinero en soles, diseñe un algoritmo que exprese dicha cantidad en euros, en marcos y en dólares. Considere los siguientes tipos de cambio.

$$1 \text{ dólar} = 3.51 \text{ soles}$$

$$1 \text{ dólar} = 1.09 \text{ euros}$$

$$1 \text{ dólar} = 2.12 \text{ marcos}$$

6. Dada la capacidad de un recipiente en galones, diseñe un algoritmo que exprese dicha capacidad en litros, en metros cúbicos y en pies cúbicos. Considere los siguientes factores de conversión:

$$1 \text{ galón} = 3.79 \text{ litros}$$

$$1 \text{ pie cúbico} = 0.0283 \text{ metros cúbicos}$$

$$1 \text{ metro cúbico} = 1000 \text{ litros}$$

7. Una persona ha recorrido tres tramos de una carretera. La longitud del primer tramo esta dada en kilómetros, del segundo tramo en pies y del tercer tramo en millas. Diseñe un programa que determine la longitud total recorrida en metros y en yardas. Considere los siguientes factores de conversión:

$$1 \text{ metro} = 3.2808 \text{ pies}$$

1 yarda = 3 pies
1 kilómetro = 1000 metros
1 milla = 1609 metros

8. En países de habla inglesa, es común dar la estatura de una persona como la suma de una cantidad entera de pies mas una cantidad entera de pulgadas. Así, la estatura de una persona podría ser 3' 2". Diseñe un programa que determine la estatura de una persona en metros conociendo su estatura en el formato inglés. Considere que:

1 pie = 12 pulgadas
1 pulgada = 2.54 centímetros
1 metro = 100 centímetros

9. Una persona tiene dos recipientes llenos de gasolina cuyas capacidades están dadas en galones y pies cúbicos, respectivamente. Diseñe un programa que determine la cantidad total de gasolina en metros cúbicos, en pies cúbicos y en yardas cúbicas. Considere los siguientes factores de conversión:

1 pie cúbico = 0.0283 metros cúbicos
1 galón = 3.79 litros
1 metro cúbico = 1000 litros
1 yarda cúbica = 27 pies cúbicos

10. En una competencia atlética el tiempo se mide en minutos, segundos y centésimas de segundo y el espacio recorrido se mide en metros. Diseñe un programa que determine la velocidad promedio de un atleta en km/hr, sabiendo que:

$\text{velocidad} = \text{espacio} / \text{tiempo}$

y que:

1 hora = 60 minutos
1 minuto = 60 segundos
1 segundo = 100 centésimas de segundo
1 kilómetro = 1000 metros

11. La repartición de ganancias en una empresa se hace en forma proporcional al número de acciones de cada uno de sus tres socios. Dada la ganancia de un año, diseñe un algoritmo que determine el monto que le corresponde a cada socio.
12. En una tienda han puesto en oferta la venta de todos sus artículos por cambio de estación ofreciendo un "15 % + 15 %" de descuento. El primer 15% se aplica al importe de la compra, mientras que el segundo 15% se aplica al importe que resulta de restar el importe de la compra menos el primer descuento. Dada la cantidad de unidades adquiridas de un mismo tipo de artículo por parte de un cliente y el precio unitario del artículo, diseñe un algoritmo que determine el importe de la compra, el importe del descuento y el importe a pagar.
13. Una tienda vende un producto cuyo costo unitario es S/. 17.5. Como oferta, la tienda ofrece un descuento fijo del 11% del importe de la compra. Adicionalmente la tienda obsequia 5 caramelos por cada docena de productos adquiridos. Diseñe un algoritmo 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.

14. Una empresa paga a sus vendedores un sueldo básico mensual de S/.300. El sueldo bruto es igual al sueldo básico más una comisión, que es igual al 9% del monto total vendido. Por ley, todo vendedor se somete a un descuento del 11%. Diseñe un programa que calcule la comisión, el sueldo bruto, el descuento y el sueldo neto de un vendedor de la empresa.
15. Diseñe un algoritmo, para una empresa de ventas de piezas de automóviles, que determine el precio al que debe vender una pieza considerando un porcentaje de ganancia. Para ello se leerán el precio de compra de la pieza y el porcentaje de ganancia que desea obtener la empresa en tanto por ciento.
16. Un club ha adquirido polos y gorras para sus socios con un descuento del 15% para los polos y de 5% para las gorras. Diseñe un algoritmo que determine, en total, el importe de la compra, el importe del descuento y el importe a pagar por la compra efectuada.
17. Diseñe un algoritmo que calcule el sueldo bruto, el descuento por ESSALUD, el descuento por AFP y el sueldo neto del empleado de una empresa de acuerdo a los siguientes criterios: el sueldo bruto se calcula multiplicando el número de horas trabajadas por una tarifa horaria, el descuento por ESSALUD es igual al 9% del sueldo bruto, el descuento por AFP es igual al 12.5% del sueldo bruto, el sueldo neto es la diferencia entre el sueldo bruto y el descuento total.
18. Un hospital ha recibido una donación especial que será repartida entre las áreas de Pediatría, Medicina General, Ginecología y Traumatología de la siguiente forma:
 - Pediatría: 20% del monto total recibido entre Medicina General y Ginecología
 - Medicina General : 45% de la donación
 - Ginecología : 30% de la donación
 - Traumatología: lo que resta la donación.

Diseñe un algoritmo que determine cuánto recibirá cada área

19. Dado un número natural de cuatro cifras, diseñe un algoritmo que forme un número con la cifra de los millares y la cifra de las unidades, en ese orden. Así, por ejemplo, si se ingresara el número **8235**, el número formado sería **85**.
20. Dado un número natural de cinco cifras, diseñe un algoritmo que forme un nuevo número intercambiando las cifras extremas del número dado. Así, por ejemplo, si se ingresara el número **14567**, el número formado sería **74561**.
21. Dado un número natural de cinco cifras, diseñe un algoritmo que elimine la cifra central. Por ejemplo, si se ingresa el número **12345**, el algoritmo deberá eliminar la cifra 3, con lo que el nuevo número es **1245**.
22. Dado un número natural de tres cifras, diseñe un algoritmo que permita obtener el revés del número. Así, si se ingresa el número **238** el revés del número es **832**.
23. Dada la hora del día en el formato HH:MM:SS, diseñe un algoritmo que determine cuanto tiempo falta para terminar el día expresado en el formato HH:MM:SS. Por ejemplo, si la hora actual del día es **15:30:20**, el tiempo que falta para terminar el día es **8:29:40**.

24. Dada una hora del día en el formato HH:MM:SS, diseñe un algoritmo que determine la hora del día luego de 200 segundos transcurridos. Por ejemplo, si la hora actual del día es 17:30:50, la hora de día luego de 200 segundos será 17:34:10.
25. Diseñe un programa para sumar dos tiempos dados en el formato HH:MM:SS. La suma también debe ser expresada de la misma forma
26. Diseñe un algoritmo que determine la duración de un viaje en el formato HH:MM:SS conociendo la horas de partida y de llegada, ambas en el formato HH:MM:SS.
27. Dada una cantidad de dinero en soles, diseñe un programa que descomponga dicha cantidad en billetes de S/. 100, S/. 50, S/.10 y monedas de S/. 5, S/. 2 y S/.1. Así, por ejemplo, S/. 3778 puede descomponerse en 37 billetes de S/. 100, mas 1 billete de S/. 50, mas 2 billetes de S/. 10, mas 1 moneda de S/. 5, mas 1 moneda de S/.2 y más 1 moneda de S/. 1.
28. Diseñe un programa que determine el porcentaje de varones y de mujeres que hay en un salón de clases.

Para recordar

- Las instrucciones secuenciales se efectúan de arriba hacia abajo, por lo que si una variable requiere de otras, las otras variables tienen que haber sido ingresadas o calculadas previamente.

UNIDAD DE
APRENDIZAJE

3

SEMANA

5

ESTRUCTURA DE SELECCIÓN SIMPLE IF

LOGRO DE LA UNIDAD DE APRENDIZAJE

Al finalizar la unidad, los alumnos, utilizando estructuras algorítmicas de selección, diseñarán algoritmos en pseudocódigo que resuelvan problemas que involucren tomas de decisiones.

TEMARIO

1. Estructura de selección simple if.
2. Operadores lógicos y relacionales.

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).

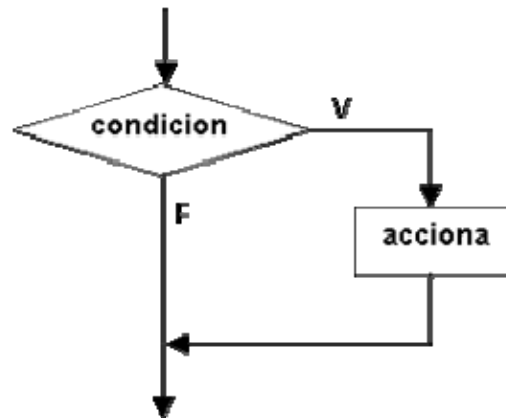


Figura 3.1 Diagrama de flujo de la estructura de selección simple if

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, estas deben estar encerradas entre llaves de bloque {}.

Código Java	Pseudocódigo
<pre>if(condicion) accionA;</pre>	<pre>si(condicion) accionA</pre>
<pre>if(condicion){ acciónA1; acciónA2; . . . accionAn; }</pre>	<pre>si(condicion){ accionA1 accionA2 . . . accionAn }</pre>

2. OPERADORES LOGICOS Y RELACIONALES

Son operadores que se utilizan para crear condiciones lógicas. Una condición lógica es una expresión lógica que puede ser verdadera (**true**) o falsa (**false**) y puede incluir operadores aritméticos.

2.1 Operadores relacionales

Los operadores relacionales se utilizan para escribir condiciones que describan la relación entre dos valores. El conjunto de operadores relacionales se muestran en la Tabla 3.1.

Tabla 3.1 Operadores relacionales

Operador	Significado
==	Igual a
!=	Diferente de
>	Mayor que
<	Menor que
>=	Mayor o igual que
<=	Menor o igual que

2.2 Operadores lógicos

Son operadores que permiten relacionar varias expresiones lógicas. El conjunto de operadores lógicos se muestra en la tabla 3.2

Tabla 3.2 Operadores lógicos

Operador	Significado
	OR lógico (ó)
&&	AND lógico (y)
!	NOT lógico (no)

Las tablas de verdad de los operadores lógicos son las mismas de la lógica matemática, como se muestra en la tabla 3.3.

Tabla 3.3 Tabla de verdad de los operadores lógicos

p	q	p && q	p q	!p
true	true	true	true	false
true	false	false	true	false
false	true	false	true	true
false	false	false	false	true

3. PROBLEMAS RESUELTOS

Problema 3. 1

Una tienda vende un producto a precios unitarios que dependen de la cantidad de unidades adquiridas de acuerdo a 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%.

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, impdes, imppag

// Entrada de datos
Leer unidades

// Cálculo del 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

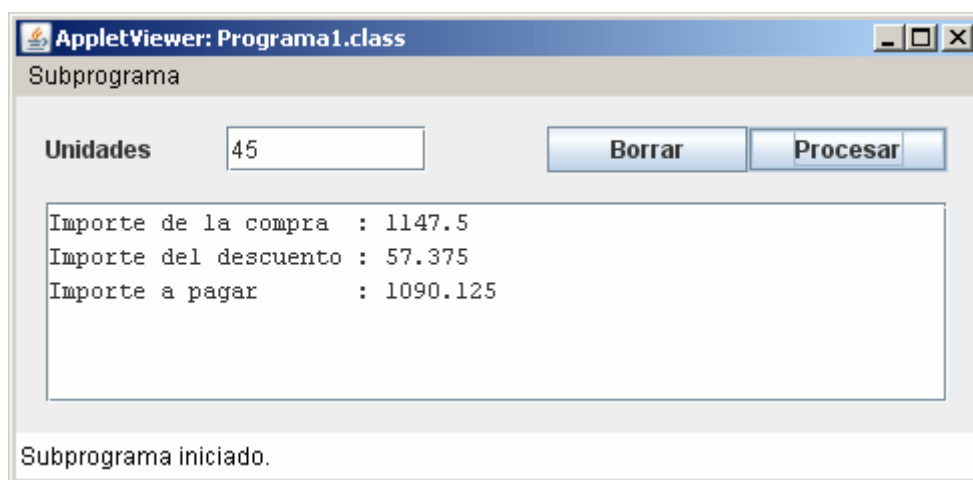
// Cálculo del importe del descuento
si( unidades > 50 )
    impdes = 0.15*impcom
si( unidades <= 50 )
    impdes = 0.05*impcom

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

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

Fin

Programa



```
import java.awt.event.*;
import java.awt.*;
import javax.swing.*;

public class Programa1 extends JApplet implements ActionListener{
    // Declaración de variables
    JButton btnProcesar, btnBorrar;
    JTextArea txtS;
```

```
JLabel lblUnidades;
JTextField txtUnidades;
JScrollPane scpScroll;

// Crea la interfaz gráfica de usuario
public void init() {
    getContentPane().setLayout(null);

    lblUnidades = new JLabel("Unidades");
    lblUnidades.setBounds(15, 15, 90, 23);
    getContentPane().add(lblUnidades);

    txtUnidades = new JTextField();
    txtUnidades.setBounds(105, 15, 100, 23);
    getContentPane().add(txtUnidades);

    btnProcesar = new JButton("Procesar");
    btnProcesar.setBounds(365, 15, 100, 23);
    btnProcesar.addActionListener(this);
    getContentPane().add(btnProcesar);

    btnBorrar = new JButton("Borrar");
    btnBorrar.setBounds(265, 15, 100, 23);
    btnBorrar.addActionListener(this);
    getContentPane().add(btnBorrar);

    txtS = new JTextArea();
    txtS.setFont(new Font("monospaced", 0, 12));

    scpScroll = new JScrollPane(txtS);
    scpScroll.setBounds(15, 53, 450, 100);
    getContentPane().add(scpScroll);
}

// Procesa eventos de tipo ActionEvent
public void actionPerformed( ActionEvent e ){

    // Si se hizo clic en Procesar
    if( e.getSource() == btnProcesar ){

        // Declaración de variables
        int unidades;
        double impcom = 0, impdes = 0, imppag;

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

        // Cálculo del 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;

        // Cálculo del importe del descuento
        if( unidades > 50 )
            impdes = 0.15*impcom;
        if( unidades <= 50 )
            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);
    }

    // Si se hizo clic en Borrar
    if( e.getSource() == btnBorrar ){
        txtUnidades.setText("");
        txtS.setText("");
        txtUnidades.requestFocus();
    }
}
}

```

Problema 3. 2

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° a 360° .

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

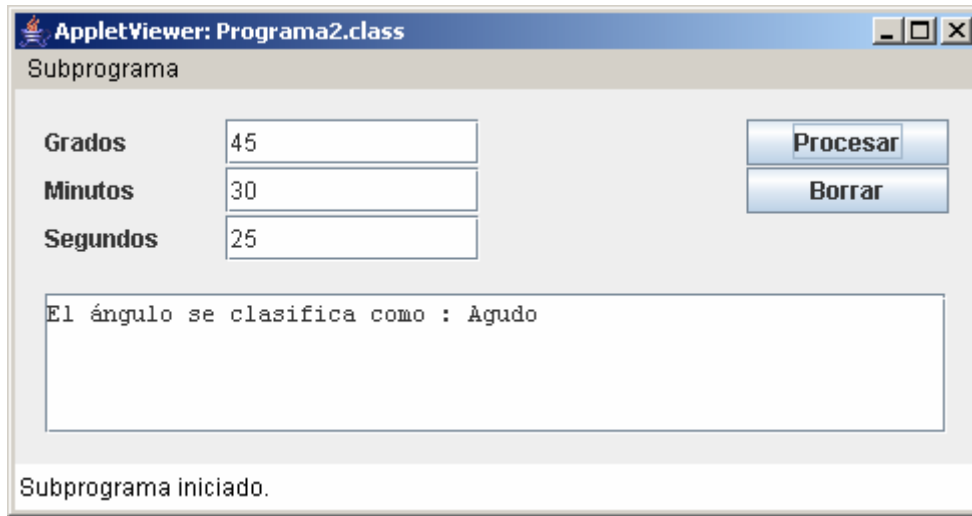
// Determina el tipo de ángulo
si( beta == 0 )
    tipo = "Nulo"
si( beta > 0 && beta < 90 )
    tipo = "Agudo"
si( beta == 90 )
    tipo = "Recto"
si( beta > 90 && beta < 180 )
    tipo = "Obtuso"
si( beta == 180 )
    tipo = "Llano"
si( beta > 180 && beta < 360 )
    tipo = "Cóncavo"
si( beta == 360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

// Salida de resultados

```

Imprimir tipo
Fin

Programa



```
import java.awt.event.*;
import java.awt.*;
import javax.swing.*;

public class Programa2 extends JApplet implements ActionListener{
    // Declaración de variables
    JButton btnBorrar, btnProcesar;
    JLabel lblGrados, lblMinutos, lblSegundos;
    JTextField txtGrados, txtMinutos, txtSegundos;
    JTextArea txtS;
    JScrollPane scpScroll;

    // Crea la interfaz gráfica de usuario GUI
    public void init() {
        getContentPane().setLayout(null);

        lblGrados = new JLabel("Grados");
        lblGrados.setBounds(15, 15, 90, 23);
        getContentPane().add(lblGrados);

        lblMinutos = new JLabel("Minutos");
        lblMinutos.setBounds(15, 39, 90, 23);
        getContentPane().add(lblMinutos);

        lblSegundos = new JLabel("Segundos");
        lblSegundos.setBounds(15, 63, 90, 23);
        getContentPane().add(lblSegundos);

        txtGrados = new JTextField();
        txtGrados.setBounds(105, 15, 127, 23);
        getContentPane().add(txtGrados);

        txtMinutos = new JTextField();
        txtMinutos.setBounds(105, 39, 127, 23);
        getContentPane().add(txtMinutos);

        txtSegundos = new JTextField();
        txtSegundos.setBounds(105, 63, 127, 23);
        getContentPane().add(txtSegundos);

        btnProcesar = new JButton("Procesar");
```

```

        btnProcesar.setBounds(365, 15, 101, 23);
        btnProcesar.addActionListener(this);
        getContentPane().add(btnProcesar);

        btnBorrar = new JButton("Borrar");
        btnBorrar.setBounds(365, 39, 101, 23);
        btnBorrar.addActionListener(this);
        getContentPane().add(btnBorrar);

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

        scpScroll = new JScrollPane(txtS);
        scpScroll.setBounds(15, 102, 450, 100);
        getContentPane().add(scpScroll);
    }

    // Procesa eventos de tipo ActionEvent
    public void actionPerformed( ActionEvent e ){

        // Si se hizo clic en Procesar
        if( e.getSource() == btnProcesar ){

            // 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);
        }

        // Si se hizo clic en Borrar
        if( e.getSource() == btnBorrar ){
            txtGrados.setText("");
            txtMinutos.setText("");
            txtSegundos.setText("");
            txtS.setText("");
            txtGrados.requestFocus();
        }
    }
}

```

Problema 3.3

El promedio final de un curso se obtiene en base al promedio simple de tres prácticas calificadas. Para ayudar a los alumnos, el profesor del curso ha prometido incrementar en dos puntos la nota de la tercera práctica calificada, si es que esta es no menor que 10. Diseñe un programa que determine el promedio final de un alumno conociendo sus tres notas. No use operadores lógicos en la solución y 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 p3
    si( p3 >= 10 ){
        p3 = p3 + 2
        si( p3 > 20 )
            p3 = 20
    }

    // Determina el promedio
    promedio = (p1+p2+p3)/3

    // Salida de resultados
    Imprimir promedio
Fin
```

Programa

```
import java.awt.event.*;
import java.awt.*;
import javax.swing.*;

public class Programa3 extends JApplet implements ActionListener{
    // Declaración de variables
    JButton btnBorrar, btnProcesar;
    JLabel lblPractica1, lblPractica2, lblPractica3;
    JTextField txtPractica1, txtPractica2, txtPractica3;
```

```
JTextArea txtS;
JScrollPane scpScroll;

// Crea la interfaz gráfica de usuario GUI
public void init() {
    getContentPane().setLayout(null);

    lblPractical1 = new JLabel("Practical");
    lblPractical1.setBounds(15, 15, 90, 23);
    getContentPane().add(lblPractical1);

    lblPractica2 = new JLabel("Practica2");
    lblPractica2.setBounds(15, 39, 90, 23);
    getContentPane().add(lblPractica2);

    lblPractica3 = new JLabel("Practica3");
    lblPractica3.setBounds(15, 63, 90, 23);
    getContentPane().add(lblPractica3);

    txtPractical1 = new JTextField();
    txtPractical1.setBounds(105, 15, 127, 23);
    getContentPane().add(txtPractical1);

    txtPractica2 = new JTextField();
    txtPractica2.setBounds(105, 39, 127, 23);
    getContentPane().add(txtPractica2);

    txtPractica3 = new JTextField();
    txtPractica3.setBounds(105, 63, 127, 23);
    getContentPane().add(txtPractica3);

    btnProcesar = new JButton("Procesar");
    btnProcesar.setBounds(365, 15, 101, 23);
    btnProcesar.addActionListener(this);
    getContentPane().add(btnProcesar);

    btnBorrar = new JButton("Borrar");
    btnBorrar.setBounds(365, 39, 101, 23);
    btnBorrar.addActionListener(this);
    getContentPane().add(btnBorrar);

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

    scpScroll = new JScrollPane(txtS);
    scpScroll.setBounds(15, 102, 450, 100);
    getContentPane().add(scpScroll);
}

// Procesa eventos de tipo ActionEvent
public void actionPerformed( ActionEvent e ){

    // Si se hizo clic en Procesar
    if( e.getSource() == btnProcesar ){

        // Declaración de variables
        double promedio;
        int p1, p2, p3;

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

        // 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);
}

// Si se hizo clic en Borrar
if( e.getSource() == btnBorrar ){
    txtPractical.setText("");
    txtPractica2.setText("");
    txtPractica3.setText("");
    txtS.setText("");
    txtPractical.requestFocus();
}
}
}

```

Problema 3.4

Diseñe un programa que lea un número natural de cuatro cifras y forme el mayor número posible de dos cifras usando la cifra mayor y la cifra menor del número ingresado.

Algoritmo

Inicio

```

// Declaración de variables
entero numero, cunid, cdece, ccent, cmill, cmay, cmen, maynum

// Entrada de datos
Leer numero

// Descompone el número en sus cifras
cmill = numero/1000
ccent = numero%1000/100
cdece = numero%1000%100/10
cunid = numero%1000%100%10

// Determina la cifra mayor
cmay = cmill
si(ccent > cmay)
    cmay = ccent
si(cdece > cmay)
    cmay = cdece
si(cunid > cmay)
    cmay = cunid

// Determina la cifra menor
cmen = cmill
si(ccent < cmen)
    cmen = ccent
si(cdece < cmen)
    cmen = cdece
si(cunid < cmen)
    cmen = cunid

// Forma el mayor número posible con las cifras mayor y menor
maynum = cmay*10+cmen

// Salida de resultados
Imprimir may, cmen, maynum

```

Fin

Programa



```
import java.awt.event.*;
import java.awt.*;
import javax.swing.*;

public class Programa4 extends JApplet implements ActionListener{
    // Declaración de variables
    JButton btnProcesar, btnBorrar;
    JTextArea txtS;
    JLabel lblNumero;
    JTextField txtNumero;
    JScrollPane scpScroll;

    // Crea la interfaz gráfica de usuario
    public void init() {
        getContentPane().setLayout(null);

        lblNumero = new JLabel("Numero");
        lblNumero.setBounds(15, 15, 90, 23);
        getContentPane().add(lblNumero);

        txtNumero = new JTextField();
        txtNumero.setBounds(105, 15, 100, 23);
        getContentPane().add(txtNumero);

        btnProcesar = new JButton("Procesar");
        btnProcesar.setBounds(365, 15, 100, 23);
        btnProcesar.addActionListener(this);
        getContentPane().add(btnProcesar);

        btnBorrar = new JButton("Borrar");
        btnBorrar.setBounds(265, 15, 100, 23);
        btnBorrar.addActionListener(this);
        getContentPane().add(btnBorrar);

        txtS = new JTextArea();
        txtS.setFont(new Font("monospaced", 0, 12));

        scpScroll = new JScrollPane(txtS);
        scpScroll.setBounds(15, 53, 450, 100);
        getContentPane().add(scpScroll);
    }

    // Procesa eventos de tipo(ActionEvent)
    public void actionPerformed( ActionEvent e ){

        // Si se hizo clic en Procesar
```

```

if( e.getSource() == btnProcesar ){

    // Declaración de variables
    int numero, cunid, cdece, ccent, cmill, cmay, cmen, maynum;

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

    // Descompone el número en sus cifras
    cmill = numero/1000;
    ccent = numero%1000/100;
    cdece = numero%1000%100/10;
    cunid = numero%1000%100%10;

    // Determina la cifra mayor
    cmay = cmill;
    if(ccent > cmay)
        cmay = ccent;
    if(cdece > cmay)
        cmay = cdece;
    if(cunid > cmay)
        cmay = cunid;

    // Determina la cifra menor
    cmen = cmill;
    if(ccent < cmen)
        cmen = ccent;
    if(cdece < cmen)
        cmen = cdece;
    if(cunid < cmen)
        cmen = cunid;

    // Forma el mayor número posible con las cifras mayor y menor
    maynum = cmay*10+cmen;

    // Salida de resultados
    txtS.setText("Cifra mayor           : " + cmay + "\n");
    txtS.append("Cifra menor           : " + cmen + "\n");
    txtS.append("Mayor número formado : " + maynum);
}

// Si se hizo clic en Borrar
if( e.getSource() == btnBorrar ){
    txtNumero.setText("");
    txtS.setText("");
    txtNumero.requestFocus();
}
}
}

```

Problema 3. 5

En un estacionamiento, se cobra S/. 2.5 por hora o fracción de hora. Dado el tiempo de estacionamiento de un vehículo expresado en el formato HH:MM, determine el importe a pagar por concepto de estacionamiento.

Algoritmo

```

Inicio
    // Declaración de variables
    entero horas, minutos
    real importe

    // Entrada de datos
    Leer horas, minutos

```

```

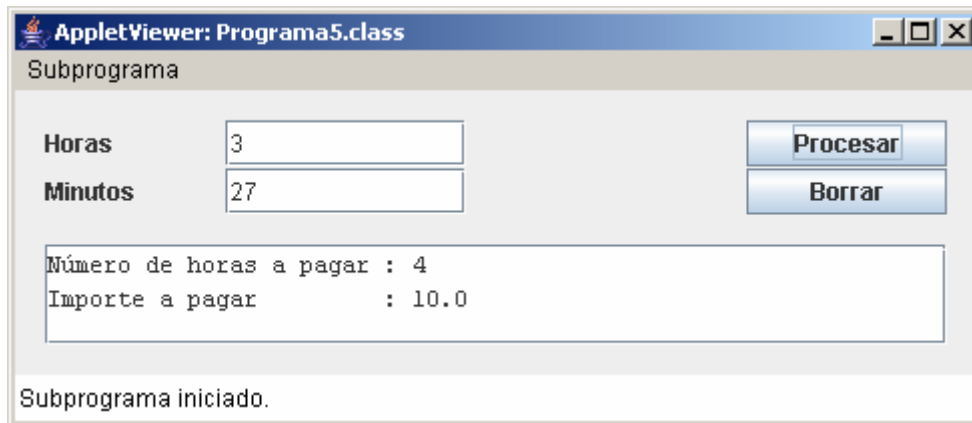
// Si hay una fracción de hora, se cobra una hora adicional
si( minutos > 0 )
    horas = horas+1

// Determina el importe a pagar
importe = horas*2.5

// Salida de resultados
Imprimir importe
Fin

```

Programa



```

import java.awt.event.*;
import java.awt.*;
import javax.swing.*;

public class Programa5 extends JApplet implements ActionListener{
    // Declaración de variables
    JButton btnProcesar, btnBorrar;
    JTextArea txtS;
    JLabel lblHoras, lblMinutos;
    JTextField txtHoras, txtMinutos;
    JScrollPane scpScroll;

    // Crea la interfaz gráfica de usuario
    public void init() {
        getContentPane().setLayout(null);

        lblHoras = new JLabel("Horas");
        lblHoras.setBounds(15, 15, 90, 23);
        getContentPane().add(lblHoras);

        lblMinutos = new JLabel("Minutos");
        lblMinutos.setBounds(15, 39, 90, 23);
        getContentPane().add(lblMinutos);

        txtHoras = new JTextField();
        txtHoras.setBounds(105, 15, 120, 23);
        getContentPane().add(txtHoras);

        txtMinutos = new JTextField();
        txtMinutos.setBounds(105, 39, 120, 23);
        getContentPane().add(txtMinutos);

        btnProcesar = new JButton("Procesar");
        btnProcesar.setBounds(365, 15, 100, 23);
        btnProcesar.addActionListener(this);
        getContentPane().add(btnProcesar);
    }
}

```

```
        btnBorrar = new JButton("Borrar");
        btnBorrar.setBounds(365, 39, 100, 23);
        btnBorrar.addActionListener(this);
        getContentPane().add(btnBorrar);

        txtS = new JTextArea();
        txtS.setFont(new Font("monospaced", 0, 12));

        scpScroll = new JScrollPane(txtS);
        scpScroll.setBounds(15, 77, 450, 50);
        getContentPane().add(scpScroll);
    }

    // Procesa eventos de tipo ActionEvent
    public void actionPerformed( ActionEvent e ){

        // Si se hizo clic en Procesar
        if( e.getSource() == btnProcesar ){

            // Declaración de variables
            int horas, minutos;
            double importe;

            // Entrada de datos
            horas = Integer.parseInt(txtHoras.getText());
            minutos = Integer.parseInt(txtMinutos.getText());

            // Si hay una fracción de hora, se cobra una hora adicional
            if( minutos > 0 )
                horas = horas+1;

            // Determina el importe a pagar
            importe = horas*2.5;

            // Salida de resultados
            txtS.setText("Número de horas a pagar : " + horas + "\n");
            txtS.append ("Importe a pagar          : " + importe);
        }

        // Si se hizo clic en Borrar
        if( e.getSource() == btnBorrar ){
            txtHoras.setText("");
            txtMinutos.setText("");
            txtS.setText("");
            txtHoras.requestFocus();
        }
    }
}
```

Problemas propuestos

Actividad

1. Dado un partido de fútbol jugado entre dos equipos A y B, diseñe un algoritmo que determine el resultado del partido entre *ganó A*, *ganó B* o *hubo empate*.
2. Una tienda vende tres tipos de productos cuyos códigos son 101, 102 y 103 a los precios unitarios dados en la siguiente tabla:

Código	Precio Unitario
101	S/. 17.5
102	S/. 25.0
103	S/. 15.5

Como oferta la tienda ofrece un porcentaje de descuento sobre el importe de la compra de acuerdo a la siguiente tabla:

Unidades adquiridas	Descuento
1 a 10	5.0%
11 a 20	7.5%
Más de 21	10.0%

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

3. Diseñe un algoritmo que determine la categoría de un estudiante en base a su promedio ponderado, de acuerdo a la siguiente tabla:

Promedio	Categoría
≥ 17	A
≥ 14 pero < 17	B
≥ 12 pero < 14	C
< 12	D

4. Diseñe un algoritmo que determine la edad menor de tres edades ingresadas.

Autoevaluación

1. Diseñe un programa que lea tres números enteros y determine el número intermedio. No use operadores lógicos en la solución.
2. Un estudiante recibe una propina mensual de S/.20. El estudiante rinde mensualmente tres exámenes (matemática, lenguaje e historia). Su papá ha decidido incentivarlo dándole una propina adicional de S/. 5 por cada examen

aprobado. Diseñe un algoritmo que determine el monto total de la propina que le corresponde al estudiante en un mes determinado.

3. En una autopista se multa a los conductores de vehiculos que exceden el límite de velocidad permitido de acuerdo a la siguiente tabla.

Velocidad (km/h)	Multa
Hasta 70	Sin sanción
71 a 90	100 euros
91 a 100	140 euros
Más de 100	200 euros

Diseñe un algoritmo que determine cuanto de multa deberá pagar un conductor.

4. Una tienda vende un producto a un precio unitario que depende del número de unidades adquiridas de acuerdo a la siguiente tabla:

Unidades adquiridas	Precio unitario
1 a 50	S/. 25.5
51 a 100	S/. 22.5
101 a 150	S/. 20.0
151 en adelante	S/. 18.0

Como oferta la tienda ofrece un descuento igual al 15% del importe de la compra si es que el número de unidades adquiridas es mayor que 50; en caso contrario, sólo descuenta el 5%.

Diseñe un algoritmo 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.

5. Un curso se evalúa en base a cuatro notas de práctica de las cuales se elimina la nota menor y se promedian las tres notas más altas. Diseñe un programa que determine la nota eliminada y el promedio final de un alumno.
6. Diseñe un algoritmo que determine el signo de un número entre positivo, negativo o cero.
7. Diseñe un programa que lea un número entero del intervalo 1 a 7, correspondiente a un día de la semana, y determine el nombre del día. Considere: 1 para lunes, 2 para martes, ..., 6 para sábado, 7 para domingo.
8. Diseñe un programa que lea un número entero del intervalo de 1 a 4, correspondiente al estado civil de una persona, y determine el nombre del estado civil. Considere: 1 para soltero, 2 para casado, 3 para viudo y 4 para divorciado.
9. ¿Que imprime el siguiente fragmento de programa?

```
int z;
z = 5;
if(z > 2);
    z = 3;
txtS.append("El valor de z es " + z);
```

10. ¿Qué imprime el siguiente fragmento de programa?

```
int a, b;
a = 8;
if(a < 20)
    b = 1;
if(a < 15)
    b = 2;
if(a < 10)
    b = 3;
if(a < 5)
    b = 4;
txtS.append("El valor de b es " + b);
```

11. ¿Qué imprimen los siguientes fragmentos de programa?

Fragmento 1

```
int a, b, c;
b = 5;
c = 1;
a = 10;
if(a > 2)
    b = 3;
    c = 2;
a = b+c;
txtS.append("El valor de a es " + a);
```

Fragmento 2

```
int a, b, c;
b = 5;
c = 1;
a = 10;
if(a > 2){
    b = 3;
    c = 2;
}
a = b+c;
txtS.append("El valor de a es " + a);
```


Para recordar

- Colocar un ; al final de la condición de un **if** hace que la acción del **if** sea nula.
- Si el cuerpo de un **if** incluye varias acciones simples, estas deben ir encerradas entre llaves de bloque { }.

UNIDAD DE
APRENDIZAJE

3

SEMANA

6

ESTRUCTURA DE SELECCIÓN DOBLE IF - ELSE

LOGRO DE LA UNIDAD DE APRENDIZAJE

Al finalizar la unidad, los alumnos, utilizando estructuras algorítmicas de selección, diseñarán algoritmos en pseudocódigo que resuelvan problemas que involucren tomas de decisiones.

TEMARIO

1. Estructura de selección doble if-else.

ACTIVIDADES

Los alumnos desarrollan algoritmos que involucren estructuras de selección if-else.

1. ESTRUCTURA DE SELECCIÓN DOBLE if - else

La *estructura de selección doble if...else (si...sino)* evalúa una condición lógica y en caso que resulte verdadera efectúa la acción **acciona**; de lo contrario, efectúa la acción **accionb**. Tanto **acciona** como **accionb** pueden ser **acciones simples** (una sola acción) o **acciones compuestas** (un bloque de acciones).

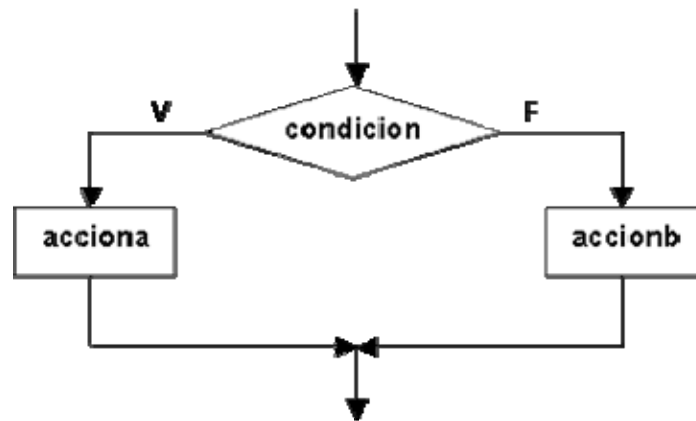


Figura 4.1 Diagrama de flujo de la estructura de selección if-else

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, estas deben estar encerradas entre llaves de bloque { }.

Código Java	Pseudocódigo
<pre> if(condicion) accionA; else accionB; </pre>	<pre> si(condicion) accionA sino accionB </pre>
<pre> if(condicion){ acciónA1; acciónA2; . . . accionAn; } else{ acciónB1; acciónB2; . . . accionBn; } </pre>	<pre> si(condicion){ acciónA1 acciónA2 . . . accionAn } sino{ acciónB1 acciónB2 . . . accionBn } </pre>

2. PROBLEMAS RESUELTOS

Problema 4.1

Una tienda han puesto en oferta la venta por docenas de cierto tipo de producto ofreciendo un descuento del 15% por la compra de no menos de 6 docenas y 10% en caso contrario. Adicionalmente la empresa ofrece un obsequio de 2 lapiceros por cada 3 docenas por la compra de no menos 30 docenas del producto; en caso contrario, no efectúa ningún obsequio. Diseñe un programa que determine el monto de la compra, el monto del descuento, el monto a pagar y la cantidad de lapiceros de obsequio por la compra de cierta cantidad de docenas del producto.

Algoritmo

Inicio

```
// Declaración de variables
entero docenas, obsequio
real montocom, montodes, montopag, precio

// Entrada de datos
Leer docenas, precio

// Calcula el monto de la compra
montocom = docenas*precio

// Calcula el monto del descuento
si (docenas >= 6)
    montodes = 0.15*montocom
sino
    montodes = 0.10*montocom

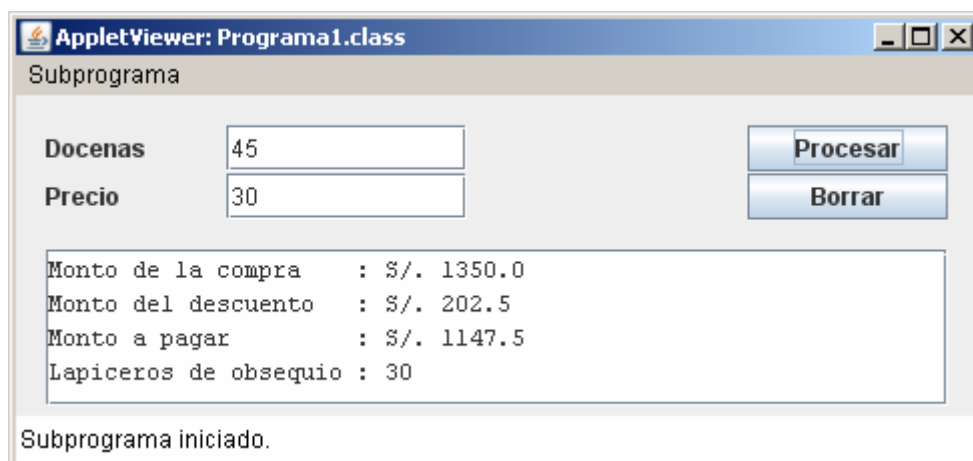
// Calcula el monto a pagar
montopag = montocom - montodes

// Calcula el número de lapiceros de obsequio
si (docenas >= 30)
    obsequio = 2*(docenas/3)
sino
    obsequio = 0

// Salida de resultados
Imprimir montocom, montodes, montopag, obsequio
```

Fin

Programa



```
import java.awt.event.*;
import java.awt.*;
import javax.swing.*;

public class Program1 extends JApplet implements ActionListener{
```

```

// Declaración de variables
JButton btnProcesar, btnBorrar;
JTextArea txtS;
JLabel lblDocenas, lblPrecio;
JTextField txtDocenas, txtPrecio;
JScrollPane scpScroll;

// Crea la interfaz gráfica de usuario
public void init() {
    getContentPane().setLayout(null);

    lblDocenas = new JLabel("Docenas");
    lblDocenas.setBounds(15, 15, 90, 23);
    getContentPane().add(lblDocenas);

    lblPrecio = new JLabel("Precio");
    lblPrecio.setBounds(15, 39, 90, 23);
    getContentPane().add(lblPrecio);

    txtDocenas = new JTextField();
    txtDocenas.setBounds(105, 15, 120, 23);
    getContentPane().add(txtDocenas);

    txtPrecio = new JTextField();
    txtPrecio.setBounds(105, 39, 120, 23);
    getContentPane().add(txtPrecio);

    btnProcesar = new JButton("Procesar");
    btnProcesar.setBounds(365, 15, 100, 23);
    btnProcesar.addActionListener(this);
    getContentPane().add(btnProcesar);

    btnBorrar = new JButton("Borrar");
    btnBorrar.setBounds(365, 39, 100, 23);
    btnBorrar.addActionListener(this);
    getContentPane().add(btnBorrar);

    txtS = new JTextArea();
    txtS.setFont(new Font("monospaced", 0, 12));

    scpScroll = new JScrollPane(txtS);
    scpScroll.setBounds(15, 77, 450, 108);
    getContentPane().add(scpScroll);
}

// Procesa eventos de tipo ActionEvent
public void actionPerformed( ActionEvent e ){

    // Si se hizo clic en Procesar
    if( e.getSource() == btnProcesar ){

        // Declaración de variables
        int docenas, obsequio;
        double montocom, montodes, montopag, precio;

        // Entrada de datos
        docenas = Integer.parseInt(txtDocenas.getText());
        precio = Double.parseDouble(txtPrecio.getText());

        // Calcula el monto de la compra
        montocom = docenas*precio;

        // Calcula el monto del descuento
        if(docenas >= 6)
            montodes = 0.15*montocom;
        else
            montodes = 0.10*montocom;
    }
}

```

```

        // Calcula el monto a pagar
        montopag = montocom - montodes;

        // Calcula el número de lapiceros de obsequio
        if(docenas >= 30)
            obsequio = 2*(docenas/3);
        else
            obsequio = 0;

        // Salida de resultados
        txtS.setText("Monto de la compra      : S/. " + montocom + "\n");
        txtS.append("Monto del descuento    : S/. " + montodes + "\n");
        txtS.append("Monto a pagar          : S/. " + montopag + "\n");
        txtS.append("Lapiceros de obsequio : " + obsequio);
    }

    // Si se hizo clic en Borrar
    if( e.getSource() == btnBorrar ){
        txtDocenas.setText("");
        txtPrecio.setText("");
        txtS.setText("");
        txtDocenas.requestFocus();
    }
}
}

```

Problema 4.2

Una institución benéfica recibe anualmente una donación proveniente de europa y lo reparte entre un centro de salud, un comedor de niños y una parte lo invierte en la bolsa de acuerdo a lo siguiente:

- Si el monto de la donación es de \$10000 o más: 30% se destina al centro de salud, 50% al comedor de niños y el resto se invierte en la bolsa.
- Si el monto de la donación es menor que \$10000: 25% se destina al centro de salud, 60% al comedor de niños y el resto se invierte en la bolsa.

La institución desea saber cuanto de dinero destinará a cada rubro anualmente.

Algoritmo

```

Inicio
    // Declaración de variables
    real donacion, comedor, salud, bolsa

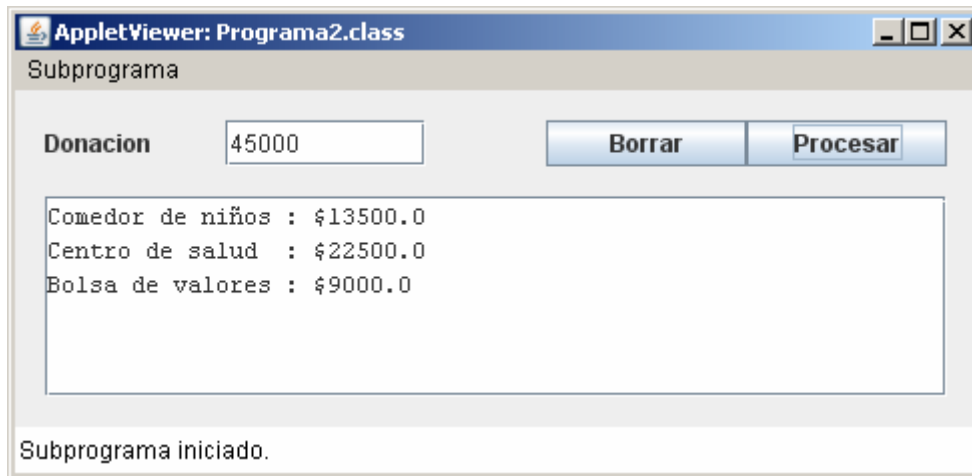
    // Entrada de datos
    Leer donacion

    // Reparte la donación
    si( donacion >= 10000 ){
        comedor = 0.30*donacion
        salud   = 0.50*donacion
        bolsa   = 0.20*donacion
    }
    sino{
        comedor = 0.25*donacion
        salud   = 0.60*donacion
        bolsa   = 0.15*donacion
    }

    // Salida de resultados
    Imprimir comedor, salud, bolsa

```

Fin

Programa

```
import java.awt.event.*;
import java.awt.*;
import javax.swing.*;

public class Programa2 extends JApplet implements ActionListener{
    // Declaración de variables
    JButton btnProcesar, btnBorrar;
    JTextArea txtS;
    JLabel lblDonacion;
    JTextField txtDonacion;
    JScrollPane scpScroll;

    // Crea la interfaz gráfica de usuario
    public void init() {
        getContentPane().setLayout(null);

        lblDonacion = new JLabel("Donacion");
        lblDonacion.setBounds(15, 15, 90, 23);
        getContentPane().add(lblDonacion);

        txtDonacion = new JTextField();
        txtDonacion.setBounds(105, 15, 100, 23);
        getContentPane().add(txtDonacion);

        btnProcesar = new JButton("Procesar");
        btnProcesar.setBounds(365, 15, 100, 23);
        btnProcesar.addActionListener(this);
        getContentPane().add(btnProcesar);

        btnBorrar = new JButton("Borrar");
        btnBorrar.setBounds(265, 15, 100, 23);
        btnBorrar.addActionListener(this);
        getContentPane().add(btnBorrar);

        txtS = new JTextArea();
        txtS.setFont(new Font("monospaced", 0, 12));

        scpScroll = new JScrollPane(txtS);
        scpScroll.setBounds(15, 53, 450, 100);
        getContentPane().add(scpScroll);
    }

    // Procesa eventos de tipo(ActionEvent)
    public void actionPerformed( ActionEvent e ){

        // Si se hizo clic en Procesar
```



```

if( e.getSource() == btnProcesar ){
    // Declaración de variables
    double donacion, comedor, salud, bolsa;

    // Entrada de datos
    donacion = Double.parseDouble(txtDonacion.getText());

    // Reparte la donación
    if( donacion >= 10000 ){
        comedor = 0.30*donacion;
        salud   = 0.50*donacion;
        bolsa   = 0.20*donacion;
    }
    else{
        comedor = 0.25*donacion;
        salud   = 0.60*donacion;
        bolsa   = 0.15*donacion;
    }

    // Salida de resultados
    txtS.setText("Comedor de niños : $" + comedor + "\n");
    txtS.append("Centro de salud : $" + salud + "\n");
    txtS.append("Bolsa de valores : $" + bolsa);
}

// Si se hizo clic en Borrar
if( e.getSource() == btnBorrar ){
    txtDonacion.setText("");
    txtS.setText("");
    txtDonacion.requestFocus();
}
}
}

```

Comentario

Note el uso de las llaves **{ }** en el bloque **if** y en el bloque **else** dado que cada bloque tiene más de una instrucción:

```

if( donacion >= 10000 ){
    comedor = 0.30*donacion;
    salud   = 0.50*donacion;
    bolsa   = 0.20*donacion;
}
else{
    comedor = 0.25*donacion;
    salud   = 0.60*donacion;
    bolsa   = 0.15*donacion;
}

```

Problema 4.3

En una oficina de empleos categorizan a los postulantes en función del sexo y de la edad de acuerdo a lo siguiente:

- Si la persona es de sexo femenino: categoría FA si tiene menos de 23 años y FB en caso contrario.
- Si la persona es de sexo masculino: categoría MA si tiene menos de 25 años y MB en caso contrario.

Diseñe un programa que determine la categoría de un postulante.

Algoritmo

```

Inicio
  // Declaración de variables
  entero sexo, edad
  cadena categoria

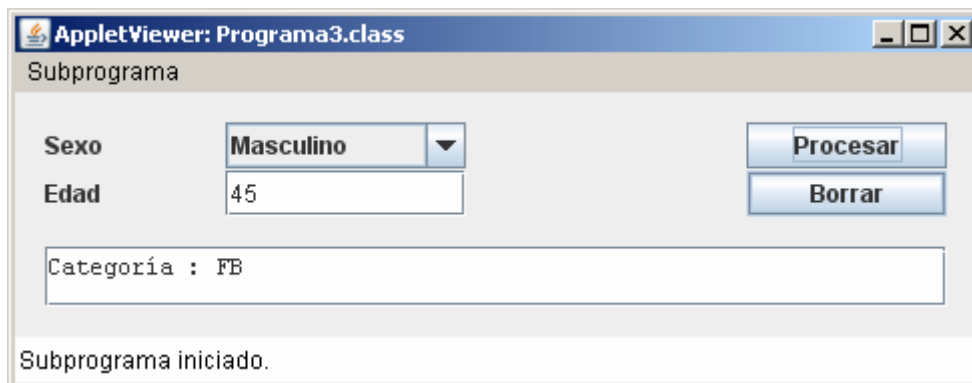
  // Entrada de datos
  Leer sexo, edad

  // Determina la categoría
  si( sexo == 0 ){
    si( edad < 23 )
      categoria = "FA"
    sino
      categoria = "FB"
  }
  sino{
    si( edad < 25 )
      categoria = "MA"
    sino
      categoria = "MB"
  }

  // Salida de resultados
  Imprimir categoria
Fin

```

Programa



```

import java.awt.event.*;
import java.awt.*;
import javax.swing.*;

public class Programa3 extends JApplet implements ActionListener{
  // Declaración de variables
  JButton btnProcesar, btnBorrar;
  JTextArea txtS;
  JLabel lblSexo, lblEdad;
  JTextField txtEdad;
  JComboBox cboSexo;
  JScrollPane scpScroll;

  // Crea la interfaz gráfica de usuario
  public void init() {
    getContentPane().setLayout(null);

    lblSexo = new JLabel("Sexo");
    lblSexo.setBounds(15, 15, 90, 23);
    getContentPane().add(lblSexo);

```

```
lblEdad = new JLabel("Edad");
lblEdad.setBounds(15, 39, 90, 23);
getContentPane().add(lblEdad);

cboSexo = new JComboBox();
cboSexo.setBounds(105, 15, 120, 23);
cboSexo.addItem("Masculino");
cboSexo.addItem("Femenino");
getContentPane().add(cboSexo);

txtEdad = new JTextField();
txtEdad.setBounds(105, 39, 120, 23);
getContentPane().add(txtEdad);

btnProcesar = new JButton("Procesar");
btnProcesar.setBounds(365, 15, 100, 23);
btnProcesar.addActionListener(this);
getContentPane().add(btnProcesar);

btnBorrar = new JButton("Borrar");
btnBorrar.setBounds(365, 39, 100, 23);
btnBorrar.addActionListener(this);
getContentPane().add(btnBorrar);

txtS = new JTextArea();
txtS.setFont(new Font("monospaced", 0, 12));

scpScroll = new JScrollPane(txtS);
scpScroll.setBounds(15, 77, 450, 30);
getContentPane().add(scpScroll);
}

// Procesa eventos de tipo ActionEvent
public void actionPerformed( ActionEvent e ){

    // Si se hizo clic en Procesar
    if( e.getSource() == btnProcesar ){

        // Declaración de variables
        int sexo, edad;
        String categoria;

        // Entrada de datos
        sexo = cboSexo.getSelectedIndex();
        edad = Integer.parseInt(txtEdad.getText());

        // Determina la categoría
        if( sexo == 0 ){
            if( edad < 23 )
                categoria = "FA";
            else
                categoria = "FB";
        }
        else{
            if( edad < 25 )
                categoria = "MA";
            else
                categoria = "MB";
        }

        // Salida de resultados
        txtS.setText("Categoría : " + categoria);
    }

    // Si se hizo clic en Borrar
    if( e.getSource() == btnBorrar ){
        txtEdad.setText("");
    }
}
```

```

        txtS.setText("");
        txtEdad.requestFocus();
    }
}

```

Problema 4.4

Diseñe un algoritmo que lea un número entero positivo tres cifras y determine si las cifras del número son o no consecutivas (en orden ascendente o en orden descendente). En caso que el número no cumpla con ser positivo de tres cifras, imprima el mensaje: "El número debe ser positivo de tres cifras".

Algoritmo

Inicio

```

// Declaración de variables
entero numero, u, d, c
cadena resultado

// Entrada de datos
Leer numero

// Prosigue si el número es correcto, sino pone un mensaje de error
si( numero >= 100 && numero <= 999 ){

    // Determina las cifras del número
    c = numero/100
    d = (numero%100)/10
    u = numero%10

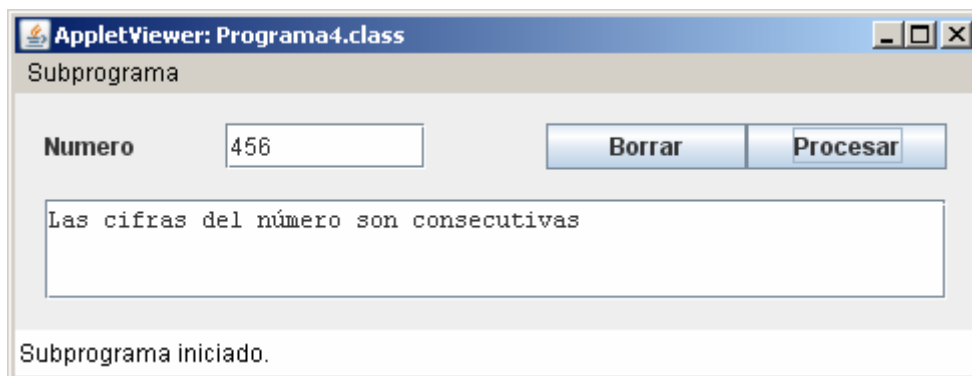
    // Determina si las cifras del número son o no consecutivas
    si((d == c+1 && u == d+1) || (d == c-1 && u == d-1))
        resultado = "Las cifras del número son consecutivas"
    sino
        resultado = "Las cifras del número no son consecutivas"

    // Salida de resultados
    Imprimir resultado
}
sino
    Imprimir "El número debe ser positivo de tres cifras"

```

Fin

Programa



```

import java.awt.event.*;
import java.awt.*;
import javax.swing.*;

```

```
public class Programa4 extends JApplet implements ActionListener{
    // Declaración de variables
    JButton btnProcesar, btnBorrar;
    JTextArea txtS;
    JLabel lblNumero;
    JTextField txtNumero;
    JScrollPane scpScroll;

    // Crea la interfaz gráfica de usuario
    public void init() {
        getContentPane().setLayout(null);

        lblNumero = new JLabel("Numero");
        lblNumero.setBounds(15, 15, 90, 23);
        getContentPane().add(lblNumero);

        txtNumero = new JTextField();
        txtNumero.setBounds(105, 15, 100, 23);
        getContentPane().add(txtNumero);

        btnProcesar = new JButton("Procesar");
        btnProcesar.setBounds(365, 15, 100, 23);
        btnProcesar.addActionListener(this);
        getContentPane().add(btnProcesar);

        btnBorrar = new JButton("Borrar");
        btnBorrar.setBounds(265, 15, 100, 23);
        btnBorrar.addActionListener(this);
        getContentPane().add(btnBorrar);

        txtS = new JTextArea();
        txtS.setFont(new Font("monospaced", 0, 12));

        scpScroll = new JScrollPane(txtS);
        scpScroll.setBounds(15, 53, 450, 50);
        getContentPane().add(scpScroll);
    }

    // Procesa eventos de tipo(ActionEvent)
    public void actionPerformed( ActionEvent e ){

        // Si se hizo clic en Procesar
        if( e.getSource() == btnProcesar ){

            // Declaración de variables
            int numero, u, d, c;
            String resultado;

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

            // Prosigue si el número es correcto, sino pone un mensaje
            // de error
            if( numero >= 100 && numero <= 999 ){

                // Determina las cifras del número
                c = numero/100;
                d = (numero%100)/10;
                u = numero%10;

                // Determina si las cifras del número son o no consecutivas
                if((d == c+1 && u == d+1) || (d == c-1 && u == d-1))
                    resultado = "Las cifras del número son consecutivas";
                else
                    resultado = "Las cifras del número no son consecutivas";

                // Salida de resultados
                txtS.setText(resultado);
            }
        }
    }
}
```

```

        }
        else
            txtS.setText("El número debe ser positivo de tres cifras");
    }

    // Si se hizo clic en Borrar
    if( e.getSource() == btnBorrar ){
        txtNumero.setText("");
        txtS.setText("");
        txtNumero.requestFocus();
    }
}
}

```

Problema 4.5

Una papelería ha puesto en oferta la venta al por mayor (en cientos) de papel bond de acuerdo a los siguientes criterios:

- Para los primeros 5 cientos, se hace un descuento del 10% por cada ciento.
- Para los cientos en exceso sobre 5, se hace un descuento del 15% por cada ciento.

Diseña un algoritmo que determine el importe bruto, el importe del descuento y el importe a pagar por una compra de papel bond.

Algoritmo

Inicio

```

// Declaración de variables
real importebru, importedes, importepag, precio
entero cientos

// Entrada de datos
Leer precio, cientos

// Determina el importe bruto
importebru = precio*cientos

// Determina el importe del descuento
si( cientos <= 5 )
    importedes = 0.10*precio*cientos
sino
    importedes = 0.10*5*precio + 0.15*(cientos-5)*precio;

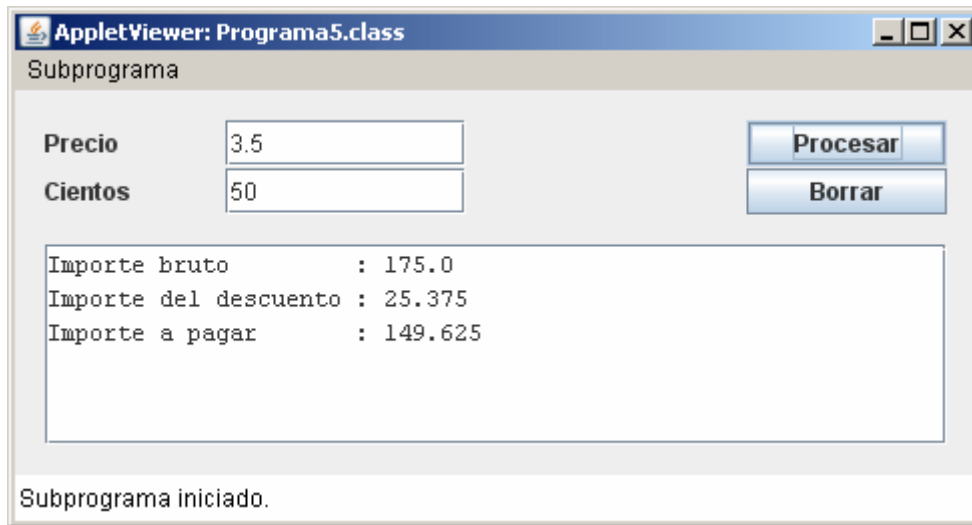
// Determina el importe a pagar
importepag = importebru - importedes

// Salida de resultados
Imprimir importebru, importedes, importepag

```

Fin

Programa



```
import java.awt.event.*;
import java.awt.*;
import javax.swing.*;

public class Programa5 extends JApplet implements ActionListener{
    // Declaración de variables
    JButton btnProcesar, btnBorrar;
    JTextArea txtS;
    JLabel lblPrecio, lblCientos;
    JTextField txtPrecio, txtCientos;
    JScrollPane scpScroll;

    // Crea la interfaz gráfica de usuario
    public void init() {
        getContentPane().setLayout(null);

        lblPrecio = new JLabel("Precio");
        lblPrecio.setBounds(15, 15, 90, 23);
        getContentPane().add(lblPrecio);

        lblCientos = new JLabel("Cientos");
        lblCientos.setBounds(15, 39, 90, 23);
        getContentPane().add(lblCientos);

        txtPrecio = new JTextField();
        txtPrecio.setBounds(105, 15, 120, 23);
        getContentPane().add(txtPrecio);

        txtCientos = new JTextField();
        txtCientos.setBounds(105, 39, 120, 23);
        getContentPane().add(txtCientos);

        btnProcesar = new JButton("Procesar");
        btnProcesar.setBounds(365, 15, 100, 23);
        btnProcesar.addActionListener(this);
        getContentPane().add(btnProcesar);

        btnBorrar = new JButton("Borrar");
        btnBorrar.setBounds(365, 39, 100, 23);
        btnBorrar.addActionListener(this);
        getContentPane().add(btnBorrar);
    }
}
```

```

txtS = new JTextArea();
txtS.setFont(new Font("monospaced", 0, 12));

scpScroll = new JScrollPane(txtS);
scpScroll.setBounds(15, 77, 450, 100);
getContentPane().add(scpScroll);
}

// Procesa eventos de tipo(ActionEvent)
public void actionPerformed(ActionEvent e){

    // Si se hizo clic en Procesar
    if( e.getSource() == btnProcesar ){

        // Declaración de variables
        double importebru, importedes, importepag, precio;
        int cientos;

        // Entrada de datos
        precio = Double.parseDouble(txtPrecio.getText());
        cientos = Integer.parseInt(txtCientos.getText());

        // Determina el importe bruto
        importebru = precio*cientos;

        // Determina el importe del descuento
        if( cientos <= 5 )
            importedes = 0.10*precio*cientos;
        else
            importedes = 0.10*5*precio + 0.15*(cientos-5)*precio;

        // Determina el importe a pagar
        importepag = importebru - importedes;

        // Salida de resultados
        txtS.setText("Importe bruto           : " + importebru + "\n");
        txtS.append("Importe del descuento : " + importedes + "\n");
        txtS.append("Importe a pagar           : " + importepag);
    }

    // Si se hizo clic en Borrar
    if( e.getSource() == btnBorrar ){
        txtPrecio.setText("");
        txtCientos.setText("");
        txtS.setText("");
        txtPrecio.requestFocus();
    }
}
}

```

Problema 4.6

Una empresa desea adquirir cierta cantidad de unidades de dos productos A y B a un proveedor cuyos precios son los siguientes:

- Producto A: costo unitario igual a S/.25.0 y 15% de descuento para más de 50 unidades adquiridas.
- Proveedor B: costo unitario igual a S/.27.5 y 10% de descuento para más de 35 unidades adquiridas.

Diseña un programa que determine el importe bruto total, el importe del descuento total y el importe total a pagar por la compra de cierta cantidad de unidades de ambos productos.

Algoritmo

```
Inicio
    // Declaración de variables
    entero unidadesA, unidadesB
    real impcomA, impdesA, imppagA
    real impcomB, impdesB, imppagB
    real impcomtot, impdestot, imppagtot

    // Entrada de datos
    Leer unidadesA, unidadesB

    // Determina los importes para el proveedor A
    impcomA = unidadesA*25.0
    si( unidadesA > 50 )
        impdesA = 0.15*impcomA
    sino
        impdesA = 0
    imppagA = impcomA - impdesA

    // Determina los importes para el proveedor B
    impcomB = unidadesB*27.5
    si( unidadesB > 35 )
        impdesB = 0.10*impcomB
    sino
        impdesB = 0
    imppagB = impcomB - impdesB

    // Determina los importes totales
    impcomtot = impcomA + impcomB
    impdestot = impdesA + impdesB
    imppagtot = imppagA + imppagB

    // Salida de resultados
    Imprimir impcomtot, impdestot, imppagtot
Fin
```

Programa

```
import java.awt.event.*;
import java.awt.*;
import javax.swing.*;
```

```

public class Programa6 extends JApplet implements ActionListener{
    // Declaración de variables
    JButton btnProcesar, btnBorrar;
    JTextArea txtS;
    JLabel lblUnidadesA, lblUnidadesB;
    JTextField txtUnidadesA, txtUnidadesB;
    JScrollPane scpScroll;

    // Crea la interfaz gráfica de usuario
    public void init() {
        getContentPane().setLayout(null);

        lblUnidadesA = new JLabel("Unidades A");
        lblUnidadesA.setBounds(15, 15, 90, 23);
        getContentPane().add(lblUnidadesA);

        lblUnidadesB = new JLabel("Unidades B");
        lblUnidadesB.setBounds(15, 39, 90, 23);
        getContentPane().add(lblUnidadesB);

        txtUnidadesA = new JTextField();
        txtUnidadesA.setBounds(105, 15, 120, 23);
        getContentPane().add(txtUnidadesA);

        txtUnidadesB = new JTextField();
        txtUnidadesB.setBounds(105, 39, 120, 23);
        getContentPane().add(txtUnidadesB);

        btnProcesar = new JButton("Procesar");
        btnProcesar.setBounds(365, 15, 100, 23);
        btnProcesar.addActionListener(this);
        getContentPane().add(btnProcesar);

        btnBorrar = new JButton("Borrar");
        btnBorrar.setBounds(365, 39, 100, 23);
        btnBorrar.addActionListener(this);
        getContentPane().add(btnBorrar);

        txtS = new JTextArea();
        txtS.setFont(new Font("monospaced", 0, 12));

        scpScroll = new JScrollPane(txtS);
        scpScroll.setBounds(15, 77, 450, 108);
        getContentPane().add(scpScroll);
    }

    // Procesa eventos de tipo ActionEvent
    public void actionPerformed( ActionEvent e ){

        // Si se hizo clic en Procesar
        if( e.getSource() == btnProcesar ){

            // Declaración de variables
            int unidadesA, unidadesB;
            double impcomA, impdesA, imppagA;
            double impcomB, impdesB, imppagB;
            double impcomtot, impdestot, imppagtot;

            // Entrada de datos
            unidadesA = Integer.parseInt(txtUnidadesA.getText());
            unidadesB = Integer.parseInt(txtUnidadesB.getText());

            // Determina los importes para el proveedor A
            impcomA = unidadesA*25.0;
            if( unidadesA > 50 )
                impdesA = 0.15*impcomA;
            else

```

```
        impdesA = 0;
        imppagA = impcomA - impdesA;

        // Determina los importes para el proveedor B
        impcomB = unidadesB*27.5;
        if( unidadesB > 35 )
            impdesB = 0.10*impcomB;
        else
            impdesB = 0;
        imppagB = impcomB - impdesB;

        // Determina los importes totales
        impcomtot = impcomA + impcomB;
        impdestot = impdesA + impdesB;
        imppagtot = imppagA + imppagB;

        // Salida de resultados
        txtS.setText("Importe de la compra total : " + impcomtot + "\n");
        txtS.append("Importe del descuento total : " + impdestot + "\n");
        txtS.append("Importe a pagar total : " + imppagtot + "\n");
    }

    // Si se hizo clic en Borrar
    if( e.getSource() == btnBorrar ){
        txtUnidadesA.setText("");
        txtUnidadesB.setText("");
        txtS.setText("");
        txtUnidadesA.requestFocus();
    }
}
}
```

Problemas propuestos

Actividad

12. En un supermercado hay una promoción según la cual el cliente raspa una tarjeta que contiene un número oculto. Si el número de la tarjeta es par no menor que 100, el cliente obtiene un descuento del 15% sobre el importe de la compra; en caso contrario, sólo se le descuenta el 5%. Dado el número oculto de la tarjeta y el importe de una compra, diseñe un algoritmo que determine el importe del descuento y el importe a pagar para un cliente del supermercado.
13. Una empresa paga a sus vendedores un sueldo bruto que es igual a la suma de un sueldo básico de S/. 250 más una comisión que es igual a un porcentaje del monto total vendido. El porcentaje por comisión depende de la categoría del vendedor de acuerdo a la siguiente tabla:

Categoría	Porcentaje por comisión
1	14.25 %
2	13.00 %
3	11.75 %

Por otro lado, si el sueldo bruto del vendedor es mayor a S/. 3500, se efectúa un descuento igual al 15% del sueldo bruto; en caso contrario, se efectúa un descuento igual al 10% del sueldo bruto.

Diseñe un algoritmo que determine el sueldo básico, la comisión, el sueldo bruto, el descuento y el sueldo neto de un vendedor de la empresa.

14. Una empresa de bienes raíces ofrece casas de interés social bajo las siguientes condiciones: si el ingreso mensual del comprador es menos de \$1250 la cuota inicial será igual al 15% del costo de la casa y el resto se distribuirá en 120 cuotas mensuales; pero, si el ingreso mensual del comprador es mayor o igual a \$1250 la cuota inicial será igual al 30% del costo de la casa y el resto se distribuirá en 75 cuotas mensuales. Diseñe un algoritmo que determine cuánto debe pagar un comprador por concepto de cuota inicial y cuánto, por cada cuota mensual.
15. Un padre ha decidido dar una propina a su hijo en base a sus notas en los cursos de Matemáticas, Física e Historia del Perú.
- Si la nota de Matemática es mayor a 17, le dará S/. 3 de propina por cada punto; en caso contrario, sólo le dará S/. 1.0 por cada punto.
 - Si la nota de Física es mayor a 15, le dará S/. 2.0 de propina por cada punto; en caso contrario, sólo le dará S/. 0.5 por cada punto.
 - Si la nota de Historia del Perú es mayor a 15, le dará S/. 1.5 por cada punto; en caso contrario, sólo le dará S/. 0.30 por cada punto.
 - Además, si la nota de Matemática es mayor a 17, le obsequiará un reloj; en caso contrario, le obsequiará un lapicero.

Diseñe un algoritmo que determine el monto total de la propina y el obsequio que le corresponde al hijo.

Autoevaluación

16. En un estacionamiento, se cobra S/. 2.5 por hora o fracción de hora. Dado el tiempo de estacionamiento de un vehículo expresado en el formato HH:MM, determine el importe a pagar por concepto de estacionamiento.
17. Una empresa paga a sus vendedores un sueldo igual al 10% del monto total vendido más S/. 25 por cada S/.500 de venta en exceso sobre S/. 5000. Diseñe un programa que permita calcular el sueldo de un vendedor.
18. Diseñe un programa que lea un número natural y determine si es o no positivo de tres cifras.
19. Una empresa ha decidido otorgar una bonificación por fiestas patrias a sus empleados. Si el empleado tiene más de un hijo, recibirá una bonificación igual al 12.5% de su sueldo bruto más S/. 40 por cada hijo; en caso contrario, solo recibirá el 12.5% de su sueldo bruto. Diseñe un programa que determine la bonificación por fiestas patrias que le corresponde a un empleado.
20. En un supermercado se hace una promoción, mediante la cual el cliente obtiene un descuento dependiendo de un número que se escoge al azar. Si el número escogido es menor que 74, el descuento es del 15% sobre el total de la compra; si es mayor o igual a 74, el descuento es del 20%. Diseñe un programa que determine cuando dinero se le debe descontar a un cliente.
21. Una persona se encuentra indecisa entre comprar un automóvil o un terreno, los cuales cuestan exactamente lo mismo. Sabe que mientras el automóvil se devalúa, con el terreno sucede lo contrario. Esta persona comprará el automóvil, si al cabo de dos años la devaluación total de este no es mayor que la mitad del incremento total del valor del terreno. Diseñe un programa que le ayude a la persona a decidir que comprar. Considere los porcentajes de devaluación del automóvil y de incremento del terreno como porcentajes anuales constantes.
22. Una empresa ha decidido adquirir varias piezas de la misma clase a una fábrica de refacciones. La empresa, dependiendo del monto total de la compra, decidirá qué hacer para pagar al fabricante. Si el monto total de la compra excede de \$500000, la empresa pedirá prestado al banco el 30% e invertirá el resto de su propio dinero; en caso contrario, pedirá prestado al banco el 20% e invertirá el resto de su propio dinero. Diseñe un programa que determine cuanto tendrá que pagar la empresa de su propio dinero y cuanto deberá pedir prestado al banco.
23. Una empresa paga a sus vendedores un sueldo bruto igual a la suma de un sueldo básico de S/. 300 más una comisión igual al 15% del monto total vendido. Por otro lado, si el sueldo bruto del vendedor es mayor que S/. 1800, recibe un descuento del 15% del sueldo bruto; en caso contrario, recibe un descuento del 11% del sueldo bruto. Además, como incentivo, la empresa obsequia 3 polos si es que el monto vendido es mayor a S/. 500; en caso contrario, sólo obsequia 1 polo. Diseñe un algoritmo que determine el sueldo bruto, el descuento, el sueldo neto y el número de polos de obsequio correspondiente a un vendedor de la empresa.

24. Diseñe un programa que lea una hora del día en formato militar y lo imprima en formato estándar. Así, si se ingresa una hora igual a 15:45:30 el programa deberá imprimir 03:05:30 P.M. En caso que se ingrese una hora inválida, imprima un mensaje de error.
25. Una empresa calcula el sueldo bruto de sus trabajadores en base a las horas trabajadas. Hasta 48 horas, se paga una tarifa horaria normal. Para las horas en exceso sobre 48, se paga un recargo del 15% respecto a la tarifa horaria normal. Por otro lado, si el sueldo bruto es superior a S/. 1700, se aplica un descuento del 11%. Diseñe un programa que determine el sueldo bruto, el descuento y el sueldo neto de un trabajador.
26. Dado un número natural de tres cifras, diseñe un algoritmo que determine si el número es o no capicúa. Un número es capicúa si se lee igual de derecha a izquierda que de izquierda a derecha. Así, por ejemplo, 363 es capicúa; pero, 356 no lo es.
27. Corrija los errores de los siguientes fragmentos de programa:

- a.

```
if(a > 20);
b = 10;
else
b = 8;
```
- b.

```
if(a+b = 5)
c = a;
d = c;
else
c = 0;
```
- c.

```
if( a > 30 );{
    if(a < 70){
        b = 5;
        c = 2;
    }
    else
        b = 1;
}
else{
    b = 0;
}
```
- d.

```
if( a > 10 & & a <= 20)
b = 5
else;
b = 2;
```

28. Determine que imprimen los siguientes fragmentos de programa cuando **a** tiene **9** y **b** tiene **11**, y cuando **a** tiene **11** y **b** tiene **9**.

- a.

```
if(a < 10)
if(b > 10)
txtS.append("*****\n");
else
txtS.append("####\n");
txtS.append("@@@@\n");
```

b.

```

if(a < 10){
    if(b > 10)
        txtS.append("*****\n");
    }
    else{
        txtS.append("####\n");
        txtS.append("@@@@\n");
    }
}

```

29. Modifique el código que sigue para producir la salida mostrada. No puede hacer ningún cambio a excepción de la inserción de llaves. Use sangrías (márgenes) para darle claridad al código, pero recuerde que las sangrías son opcionales.

```

if( a == 8 )
if( b == 5 )
    txtS.append("@@@@\n");
else
    txtS.append("####\n");
    txtS.append("$$$$\n");
    txtS.append("&&&\n");

```

- a. Para **a** igual a **5** y **b** igual a **8** debe mostrarse la siguiente salida

```

@@@@
$$$$
&&&

```

- b. Para **a** igual a **5** y **b** igual a **8** debe mostrarse la siguiente salida

```

@@@@

```

- c. Para **a** igual a **5** y **b** igual a **8** debe mostrarse la siguiente salida

```

@@@@
&&&

```

- d. Para **a** igual a **5** y **b** igual a **7** debe mostrarse la siguiente salida

```

####
$$$$
&&&

```

Para recordar

- Poner ; al final de la condición de un **if...else** causa un error de sintaxis.
- Si el cuerpo del **if** o el cuerpo del **else** incluyen más de una acción, estas deben ir encerradas entre llaves de bloque { }.

UNIDAD DE
APRENDIZAJE

3

SEMANA

8

ESTRUCTURA DE SELECCIÓN DOBLE ENCADENADA IF – ELSE - IF

LOGRO DE LA UNIDAD DE APRENDIZAJE

Al finalizar la unidad, los alumnos, utilizando estructuras algorítmicas de selección, diseñarán algoritmos en pseudocódigo que resuelvan problemas que involucren tomas de decisiones.

TEMARIO

1. La Estructura de selección doble encadenada if-else-if.

ACTIVIDADES

Los alumnos desarrollan algoritmos que involucren la estructura if-else-if.

1. LA ESTRUCTURA DE SELECCIÓN DOBLE ENCADENADA if-else-if

La *estructura de selección doble encadenada if...else...if* evalúa un conjunto de condiciones en orden descendente, pasando de una condición a otra siempre que la condición anterior sea falsa, y en el momento que encuentre una condición verdadera, efectúa la acción correspondiente a dicha condición y abandona el resto de la estructura. La estructura tiene una acción por defecto que se efectúa en el caso que todas las condiciones sean falsas.

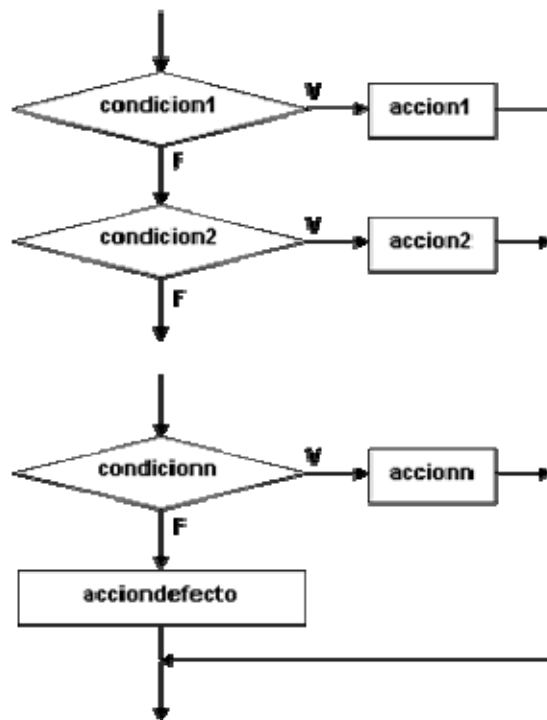


Figura 5.1 Estructura de selección doble en cascada

En las tablas que siguen se muestran el código Java y el pseudocódigo correspondientes a los dos formatos de escritura de la estructura if-else-if

Estructura de Selección if – else –if (Formato 1)

Código Java	Pseudocódigo
<pre> if(condicion1) accion1; else if(condicion2) accion2; else if(condicion3) accion3; . . . else if(condicionnn) accionnn; else acciondefecto; </pre>	<pre> si(condicion1) accion1 sino si(condicion2) accion2 sino si(condicion3) accion3 . . . sino si(condicionnn) accionnn sino acciondefecto </pre>

Estructura de Selección if – else –if (Formato 2)

Código Java	Pseudocódigo
<pre> if(condicion1) accion1; else if(condicion2) accion2; else if(condicion3) accion3; . . . else if(condicionn) accionn; else acciondefecto; </pre>	<pre> si(condicion1) accion1 sino si(condicion2) accion2 sino si(condicion3) accion3 . . . sino si(condicionn) accionn sino acciondefecto </pre>

En el caso de acciones compuestas, estas deben estar encerradas entre llaves de bloque {}.

2. PROBLEMAS RESUELTOS**Problema 5.1**

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° a 360° .

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"
sino si(beta < 90 )

```

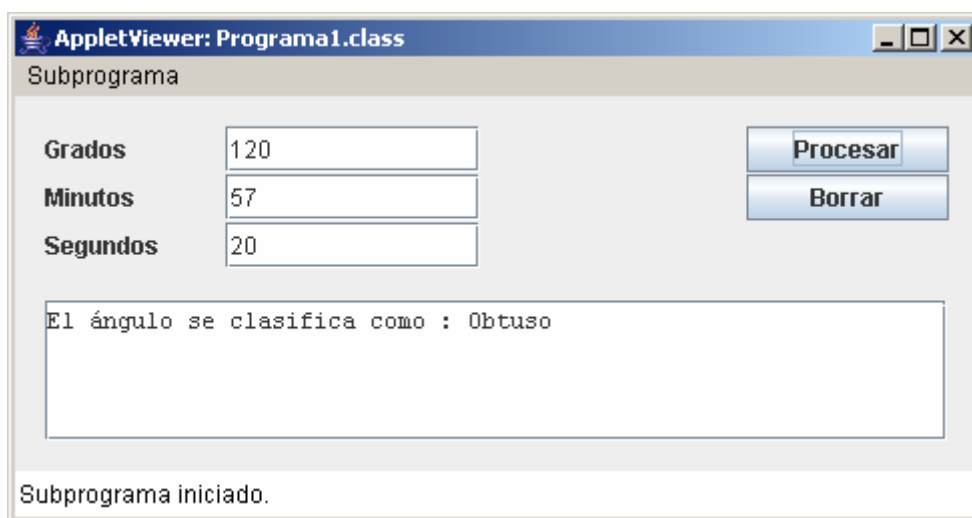
```

        tipo = "Agudo"
    sino si( beta == 90 )
        tipo = "Recto"
    sino si( beta < 180 )
        tipo = "Obtuso"
    sino si( beta == 180 )
        tipo = "Llano"
    sino si( beta < 360 )
        tipo = "Cóncavo"
    sino
        tipo = "Completo"

    // Salida de resultados
    Imprimir tipo
Fin

```

Programa



```

import java.awt.event.*;
import java.awt.*;
import javax.swing.*;

public class Program1 extends JApplet implements ActionListener{
    // Declaración de variables
    JButton btnBorrar, btnProcesar;
    JLabel lblGrados, lblMinutos, lblSegundos;
    JTextField txtGrados, txtMinutos, txtSegundos;
    JTextArea txtS;
    JScrollPane scpScroll;

    // Crea la interfaz gráfica de usuario GUI
    public void init() {
        getContentPane().setLayout(null);

        lblGrados = new JLabel("Grados");
        lblGrados.setBounds(15, 15, 90, 23);
        getContentPane().add(lblGrados);

        lblMinutos = new JLabel("Minutos");
        lblMinutos.setBounds(15, 39, 90, 23);
        getContentPane().add(lblMinutos);

        lblSegundos = new JLabel("Segundos");
        lblSegundos.setBounds(15, 63, 90, 23);
        getContentPane().add(lblSegundos);
    }
}

```

```

txtGrados = new JTextField();
txtGrados.setBounds(105, 15, 127, 23);
getContentPane().add(txtGrados);

txtMinutos = new JTextField();
txtMinutos.setBounds(105, 39, 127, 23);
getContentPane().add(txtMinutos);

txtSegundos = new JTextField();
txtSegundos.setBounds(105, 63, 127, 23);
getContentPane().add(txtSegundos);

btnProcesar = new JButton("Procesar");
btnProcesar.setBounds(365, 15, 101, 23);
btnProcesar.addActionListener(this);
getContentPane().add(btnProcesar);

btnBorrar = new JButton("Borrar");
btnBorrar.setBounds(365, 39, 101, 23);
btnBorrar.addActionListener(this);
getContentPane().add(btnBorrar);

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

scpScroll = new JScrollPane(txtS);
scpScroll.setBounds(15, 102, 450, 70);
getContentPane().add(scpScroll);
}

// Procesa eventos de tipo ActionEvent
public void actionPerformed( ActionEvent e ){

    // Si se hizo clic en Procesar
    if( e.getSource() == btnProcesar ){

        // 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";
        else if( beta < 90 )
            tipo = "Agudo";
        else if( beta == 90 )
            tipo = "Recto";
        else if( beta < 180 )
            tipo = "Obtuso";
        else if( beta == 180 )
            tipo = "Ilano";
        else if( beta < 360 )
            tipo = "Cóncono";
        else
            tipo = "Completo";

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

```

```

// Si se hizo clic en Borrar
if( e.getSource() == btnBorrar ){
    txtGrados.setText("");
    txtMinutos.setText("");
    txtSegundos.setText("");
    txtS.setText("");
    txtGrados.requestFocus();
}
}
}

```

Problema 5.2

En la playa de estacionamiento de un centro comercial, cobran una tarifa fija por hora o fracción de acuerdo a la siguiente tabla:

Día	Tarifa
Lunes a Jueves	S/. 3.5
Viernes y Sábado	S/. 4.5
Domingo	S/. 2.5

Cuando un vehículo entra al estacionamiento, se anota la hora de entrada y al retirarse, se toma nota de la hora de salida, ambas en horas y minutos. Diseñe un programa que determine cuanto debe pagar un cliente por el estacionamiento de su vehículo.

Algoritmo

Considerando: día: 0 → Lunes, 1 → Martes, ..., 6 → Domingo

Inicio

```

// Declaración de variables
entero h1, m1, h2, m2, dia, horaspag, totalmin
real montopag

// Entrada de datos
Leer h1, m1, h2, m2, dia

// Cálculo de la cantidad de horas a pagar
totalmin = (h2-h1)*60 + (m2-m1)
horaspag = totalmin/60
si( totalmin%60 != 0 )
    horaspag = horaspag+1

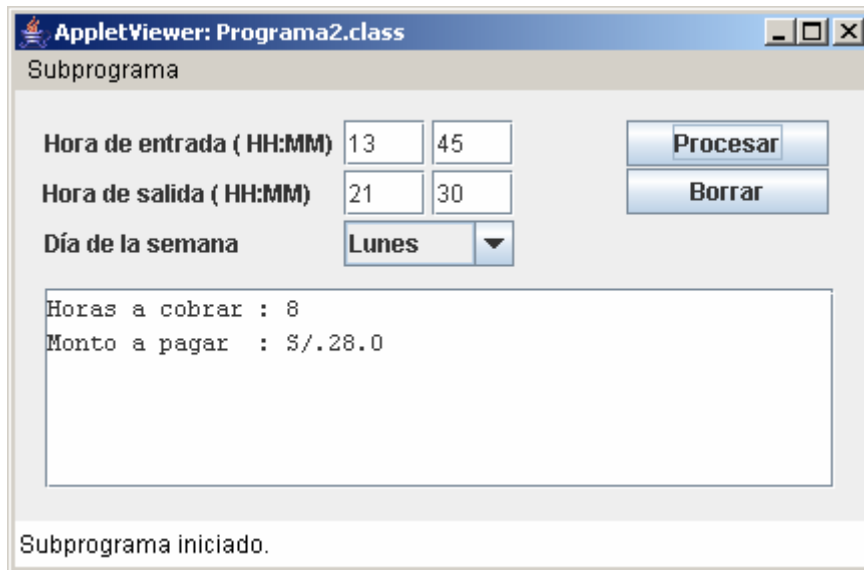
// Cálculo del monto a pagar
si( dia <= 3 )
    montopag = horaspag*3.5
sino si( dia <= 5 )
    montopag = 4.5*horaspag
sino
    montopag = 2.5*horaspag;

// Salida de resultados
Imprimir horaspag, montopag

```

Fin

Programa



```
import java.awt.event.*;
import java.awt.*;
import javax.swing.*;

public class Programa2 extends JApplet implements ActionListener{
    // Declaración de variables
    JButton btnProcesar, btnBorrar;
    JTextArea txtS;
    JComboBox choDia;
    JLabel lblDia, lblEntrada, lblSalida;
    JTextField txtH1, txtH2, txtM1, txtM2;
    JScrollPane scpScroll;

    // Crea la interfaz gráfica de usuario GUI
    public void init() {
        getContentPane().setLayout(null);

        lblDia = new JLabel("Día de la semana");
        lblDia.setBounds(15, 65, 150, 23);
        getContentPane().add(lblDia);

        lblEntrada = new JLabel("Hora de entrada ( HH:MM)");
        lblEntrada.setBounds(15, 15, 150, 23);
        getContentPane().add(lblEntrada);

        lblSalida = new JLabel("Hora de salida ( HH:MM)");
        lblSalida.setBounds(14, 40, 150, 23);
        getContentPane().add(lblSalida);

        txtH1 = new JTextField();
        txtH1.setBounds(164, 15, 41, 23);
        getContentPane().add(txtH1);

        txtM1 = new JTextField();
        txtM1.setBounds(208, 15, 41, 23);
        getContentPane().add(txtM1);

        txtH2 = new JTextField();
        txtH2.setBounds(164, 40, 41, 23);
        getContentPane().add(txtH2);

        txtM2 = new JTextField();
        txtM2.setBounds(208, 40, 41, 23);
        getContentPane().add(txtM2);

        btnProcesar = new JButton("Procesar");
```

```

        btnProcesar.setBounds(305, 15, 101, 23);
        btnProcesar.addActionListener(this);
        getContentPane().add(btnProcesar);

        btnBorrar = new JButton("Borrar");
        btnBorrar.setBounds(305, 39, 101, 23);
        btnBorrar.addActionListener(this);
        getContentPane().add(btnBorrar);

        choDia = new JComboBox();
        choDia.setBounds(164, 65, 85, 23);
        choDia.addItem("Lunes");
        choDia.addItem("Martes");
        choDia.addItem("Miércoles");
        choDia.addItem("Jueves");
        choDia.addItem("Viernes");
        choDia.addItem("Sábado");
        choDia.addItem("Domingo");
        getContentPane().add(choDia);

        txtS = new JTextArea();
        txtS.setFont(new Font("monospaced", 0, 12));

        scpScroll = new JScrollPane(txtS);
        scpScroll.setBounds(15, 99, 394, 100);
        getContentPane().add(scpScroll);
    }

    // Procesa eventos de tipo ActionEvent
    public void actionPerformed( ActionEvent e ){

        // Si se hizo clic en Procesar
        if( e.getSource() == btnProcesar ){

            // Declaración de variables
            int h1, m1, h2, m2, dia, horaspag, totalmin;
            double montopag;

            // Entrada de datos
            h1 = Integer.parseInt(txtH1.getText());
            m1 = Integer.parseInt(txtM1.getText());
            h2 = Integer.parseInt(txtH2.getText());
            m2 = Integer.parseInt(txtM2.getText());
            dia = choDia.getSelectedIndex();

            // Determina la cantidad de horas a pagar
            totalmin = (h2-h1)*60 + (m2-m1);
            horaspag = totalmin/60;
            if( totalmin%60 != 0 )
                horaspag = horaspag+1;

            // Determina el monto a pagar
            if( dia <= 3 )
                montopag = horaspag*3.5;
            else if( dia <= 5 )
                montopag = 4.5*horaspag;
            else
                montopag = 2.5*horaspag;

            // Salida de resultados
            txtS.setText("Horas a cobrar : " + horaspag + "\n");
            txtS.append("Monto a pagar : S/." + montopag);
        }

        // Si se hizo clic en Borrar
        if( e.getSource() == btnBorrar ){
            txtH1.setText("");
            txtM1.setText("");
        }
    }

```



```

        txtH2.setText("");
        txtM2.setText("");
        txtS.setText("");
        txtH1.requestFocus();
    }
}
}

```

Problema 5.3

En una universidad, los alumnos están categorizados en cuatro categorías. A cada categoría le corresponde una pensión mensual distinta dada en la siguiente tabla:

Categoría	Pensión
A	S/. 550
B	S/. 500
C	S/. 460
D	S/. 400

Semestralmente, la universidad efectúa rebajas en las pensiones de sus estudiantes a partir del segundo ciclo en base al promedio ponderado del ciclo anterior en porcentajes dados en la tabla siguiente:

Promedio	Descuento
0 a 13.99	No hay descuento
14.00 a 15.99	10 %
16.00 a 17.99	12 %
18.00 a 20.00	15 %

Diseñe un programa que determine cuánto de rebaja recibirá un estudiante sobre su pensión actual y a cuánto asciende su nueva pensión.

Algoritmo

```

Inicio
    // Declaración de variables
    entero categoria
    real actualpen, nuevapen, descuento, promedio

    // Entrada de datos
    Leer categoria, promedio

    // Cálculo de la pensión actual
    si( categoria == 0 )
        actualpen = 550
    sino si ( categoria == 1 )
        actualpen = 500
    sino si ( categoria == 2 )
        actualpen = 460
    sino
        actualpen = 400

    // Cálculo del descuento
    si( promedio <= 13.99 )
        descuento = 0
    sino si( promedio <= 15.99 )
        descuento = 0.10*actualpen
    sino si( promedio <= 17.99 )
        descuento = 0.12*actualpen

```

```

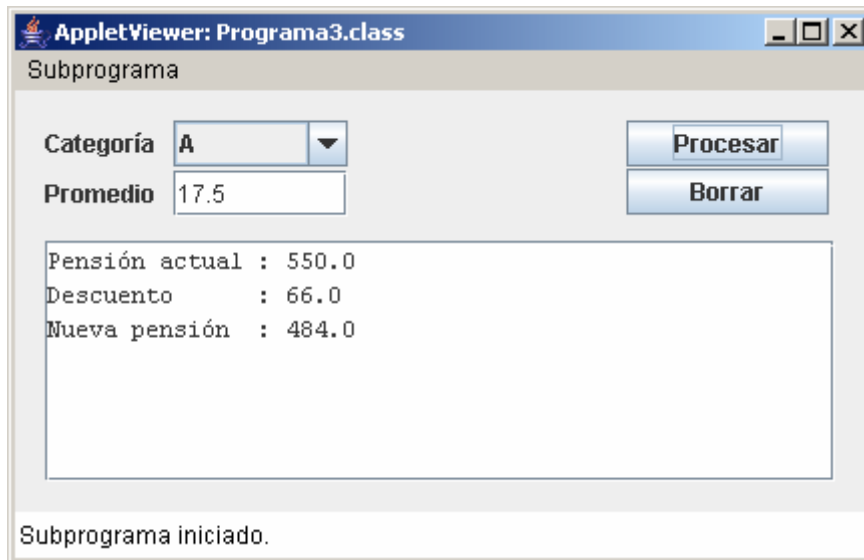
    sino
        descuento = 0.15*actualpen

    // Cálculo de la nueva pensión
    nuevapen = actualpen - descuento

    // Salida de resultados
    Imprimir actualpen, nuevapen
Fin

```

Programa



```

import java.awt.event.*;
import java.awt.*;
import javax.swing.*;

public class Programa3 extends JApplet implements ActionListener{
    // Declaración de variables
    JButton btnProcesar, btnBorrar;
    JTextArea txtS;
    JComboBox choCategoria;
    JLabel lblCategoria, lblPromedio;
    JTextField txtPromedio;
    JScrollPane scpScroll;

    // Crea la interfaz gráfica de usuario GUI
    public void init() {
        getContentPane().setLayout(null);
        setBackground(new Color(214,211,206));

        txtPromedio = new JTextField();
        txtPromedio.setBounds(79, 40, 87, 23);
        getContentPane().add(txtPromedio);

        btnProcesar = new JButton("Procesar");
        btnProcesar.setBounds(305, 15, 101, 23);
        btnProcesar.addActionListener(this);
        getContentPane().add(btnProcesar);

        btnBorrar = new JButton("Borrar");
        btnBorrar.setBounds(305, 39, 101, 23);
        btnBorrar.addActionListener(this);
        getContentPane().add(btnBorrar);

        choCategoria = new JComboBox();
    }
}

```

```

        choCategoria.setBounds(79, 15, 87, 23);
        choCategoria.addItem("A");
        choCategoria.addItem("B");
        choCategoria.addItem("C");
        choCategoria.addItem("D");
        getContentPane().add(choCategoria);

        lblCategoria = new JLabel("Categoría");
        lblCategoria.setBounds(15, 15, 61, 23);
        getContentPane().add(lblCategoria);

        lblPromedio = new JLabel("Promedio");
        lblPromedio.setBounds(15, 40, 56, 23);
        getContentPane().add(lblPromedio);

        txtS = new JTextArea();
        txtS.setFont(new Font("monospaced", 0, 12));

        scpScroll = new JScrollPane(txtS);
        scpScroll.setBounds(15, 75, 394, 55);
        getContentPane().add(scpScroll);
    }

    // Procesa eventos de tipo ActionEvent
    public void actionPerformed( ActionEvent e ){

        // Si se hizo clic en Procesar
        if( e.getSource() == btnProcesar ){

            // Declaración de variables
            int categoria;
            double actualpen, nuevapen, descuento, promedio;

            // Entrada de datos
            categoria = choCategoria.getSelectedIndex();
            promedio = Double.parseDouble(txtPromedio.getText());

            // Cálculo de la pensión actual
            if( categoria == 0 )
                actualpen = 550;
            else if( categoria == 1 )
                actualpen = 500;
            else if( categoria == 2 )
                actualpen = 460;
            else
                actualpen = 400;

            // Cálculo del descuento
            if( promedio <= 13.99 )
                descuento = 0;
            else if( promedio <= 15.99 )
                descuento = 0.10*actualpen;
            else if( promedio <= 17.99 )
                descuento = 0.12*actualpen;
            else
                descuento = 0.15*actualpen;

            // Cálculo de la nueva pensión
            nuevapen = actualpen - descuento;

            // Salida de resultados
            txtS.setText("Pensión actual : " + actualpen + "\n");
            txtS.append ("Descuento      : " + descuento + "\n");
            txtS.append ("Nueva pensión   : " + nuevapen);
        }

        // Si se hizo clic en Borrar
        if( e.getSource() == btnBorrar ){

```

```

        txtS.setText("");
        txtPromedio.setText("");
        txtPromedio.requestFocus();
    }
}
}

```

Problema 5.4

Una empresa de préstamos tiene el siguiente esquema de cobros:

Monto del préstamo (S/.)	Número de cuotas
Hasta 5000	2
Más de 5000 hasta 10000	4
Más de 10000 hasta 15000	6
Más de 15000	10

Si el monto del préstamo es mayor a S/. 10000, la empresa cobra 3% de interés mensual; en caso contrario, cobra 5% de interés mensual.

Dado el monto del préstamo de un cliente, diseñe un programa que determine el monto de la cuota mensual y el monto del interés total entre todas las cuotas.

Algoritmo

Inicio

```

// Declaración de variables
real montoprestamo, montointeres, tasainteres, montocuota
entero cuotas

// Entrada de datos
Leer montoprestamo

// Obtención del número de cuotas
si( montoprestamo <= 5000 )
    cuotas = 2
sino si( montoprestamo <= 10000 )
    cuotas = 4
sino si( montoprestamo <= 15000 )
    cuotas = 6
sino
    cuotas = 10

// Obtención de la tasa de interés
si( montoprestamo > 10000 )
    tasainteres = 0.03
sino
    tasainteres = 0.05

// Cálculo del monto del interés total
montointeres = tasainteres*montoprestamo*cuotas

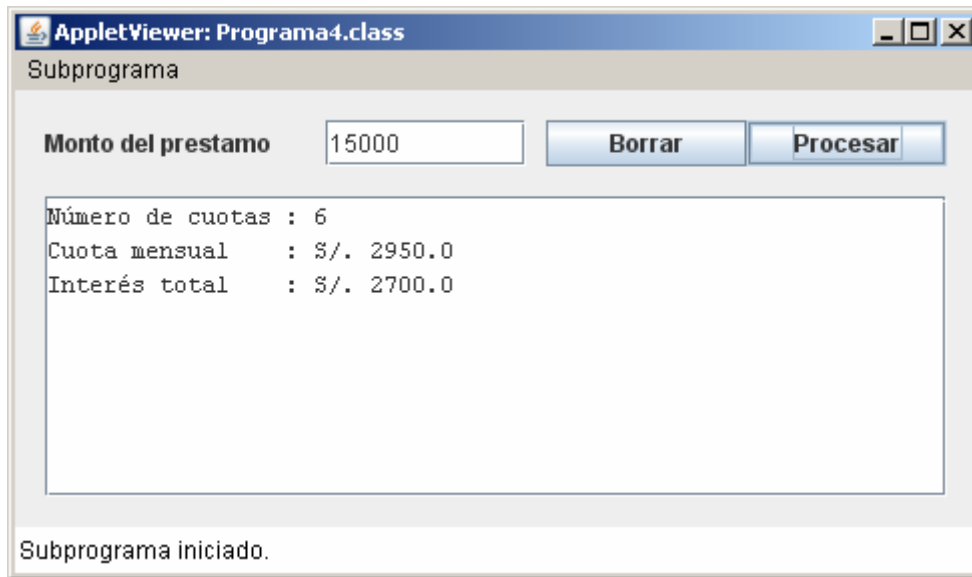
// Cálculo del monto de la cuota
montocuota = (montoprestamo + montointeres)/cuotas

// Salida de resultados
Imprimir cuotas, montocuota, montointeres

```

Inicio

Programa



```
import java.awt.event.*;
import java.awt.*;
import javax.swing.*;

public class Programa4 extends JApplet implements ActionListener{
    // Declaración de variables
    JButton btnProcesar, btnBorrar;
    JTextArea txtS;
    JLabel lblMontoPrestamo;
    JTextField txtMontoPrestamo;
    JScrollPane scpScroll;

    // Crea la interfaz gráfica de usuario
    public void init() {
        getContentPane().setLayout(null);

        lblMontoPrestamo = new JLabel("Monto del prestamo");
        lblMontoPrestamo.setBounds(15, 15, 140, 23);
        getContentPane().add(lblMontoPrestamo);

        txtMontoPrestamo = new JTextField();
        txtMontoPrestamo.setBounds(155, 15, 100, 23);
        getContentPane().add(txtMontoPrestamo);

        btnProcesar = new JButton("Procesar");
        btnProcesar.setBounds(365, 15, 100, 23);
        btnProcesar.addActionListener(this);
        getContentPane().add(btnProcesar);

        btnBorrar = new JButton("Borrar");
        btnBorrar.setBounds(265, 15, 100, 23);
        btnBorrar.addActionListener(this);
        getContentPane().add(btnBorrar);

        txtS = new JTextArea();
        txtS.setFont(new Font("monospaced", 0, 12));

        scpScroll = new JScrollPane(txtS);
        scpScroll.setBounds(15, 53, 450, 150);
        getContentPane().add(scpScroll);
    }

    // Procesa eventos de tipo(ActionEvent)
    public void actionPerformed( ActionEvent e ){
        // Si se hizo clic en Procesar
        if( e.getSource() == btnProcesar ){
```

```

// Declaración de variables
double montoprestamo, montointeres, tasainteres, montocuota;
int cuotas;

// Entrada de datos
montoprestamo = Double.parseDouble(txtMontoPrestamo.getText());

// Obtención del número de cuotas
if( montoprestamo <= 5000 )
    cuotas = 2;
else if( montoprestamo <= 10000 )
    cuotas = 4;
else if( montoprestamo <= 15000 )
    cuotas = 6;
else
    cuotas = 10;

// Obtención de la tasa de interés
if( montoprestamo > 10000 )
    tasainteres = 0.03;
else
    tasainteres = 0.05;

// Cálculo del monto del interés total
montointeres = tasainteres*montoprestamo*cuotas;

// Cálculo del monto de la cuota
montocuota = (montoprestamo + montointeres)/cuotas;

// Salida de resultados
txtS.setText("Número de cuotas : " + cuotas + "\n");
txtS.append("Cuota mensual      : S/. " + montocuota + "\n");
txtS.append("Interés total      : S/. " + montointeres + "\n");
}

// Si se hizo clic en Borrar
if( e.getSource() == btnBorrar ){
    txtMontoPrestamo.setText("");
    txtS.setText("");
    txtMontoPrestamo.requestFocus();
}
}
}

```

Problema 5.5

Una empresa evalúa a sus empleados bajo dos criterios: puntualidad y rendimiento. En cada caso el empleado recibe un puntaje que va de 1 a 10, de acuerdo a los siguientes criterios:

Puntaje por puntualidad:- está en función a los minutos de tardanza de acuerdo a la siguiente tabla:

Minutos de tardanza	Puntaje
0	10
1 a 2	8
3 a 5	6
6 a 9	4
Más de 9	0

Puntaje por rendimiento:- está en función a la cantidad de observaciones efectuadas al empleado por no cumplir sus obligaciones de acuerdo a la siguiente tabla:

Observaciones efectuadas	Puntaje
0	10
1	8
2	5
3	1
Más de 3	0

El puntaje total del empleado es la suma del puntaje por puntualidad más el puntaje por rendimiento. En base al puntaje total, el empleado recibe una bonificación anual de acuerdo a la siguiente tabla:

Puntaje total	Bonificación
Menos de 11	S/. 2.5 por punto
11 a 13	S/. 5.0 por punto
14 a 16	S/. 7.5 por punto
17 a 19	S/. 10.0 por punto
20	S/. 12.5 por punto

Diseñe un programa que determine el puntaje por puntualidad, el puntaje por rendimiento, el puntaje total y la bonificación que le corresponden a un empleado de la empresa.

Algoritmo

Inicio

```
// Declaración de variables
entero minutosTar, numeroObs, puntajePun, puntajeRen, puntajeTot
real bonificacion

// Entrada de datos
Leer minutosTar, numeroObs

// Determinación del puntaje por puntualidad
si (minutosTar == 0)
    puntajePun = 10
sino si (minutosTar <= 2)
    puntajePun = 8
sino si (minutosTar <= 5)
    puntajePun = 6
sino si (minutosTar <= 9)
    puntajePun = 4
sino
    puntajePun = 0

// Determinación del puntaje por rendimiento
si (numeroObs == 0)
    puntajeRen = 10
sino si (numeroObs == 1)
    puntajeRen = 8
sino si (numeroObs == 2)
    puntajeRen = 5
sino si (numeroObs == 3)
    puntajeRen = 1
sino
    puntajeRen = 0
```

```

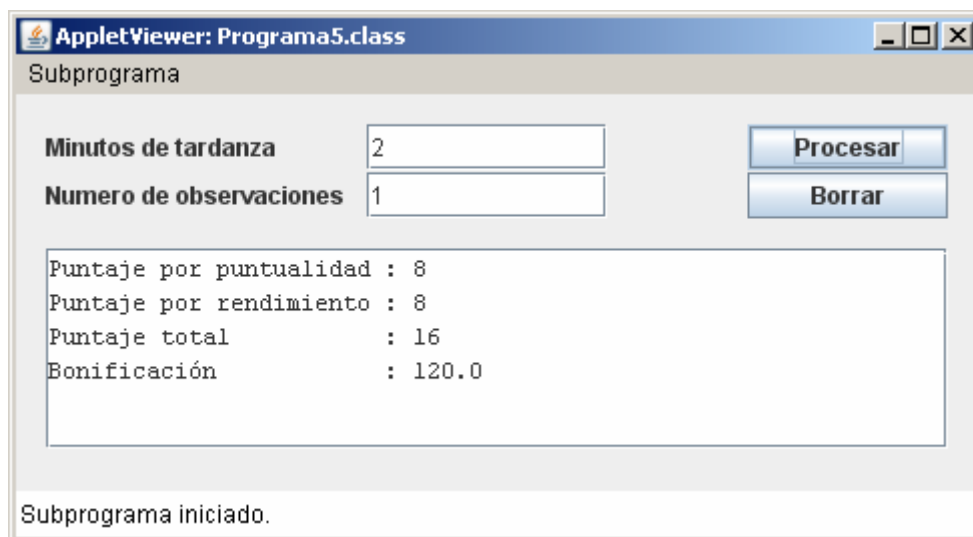
// Determinación del puntaje total
puntajeTot = puntajePun + puntajeRen

// Determinación de la bonificación
si (puntajeTot < 11)
    bonificacion = 2.5*puntajeTot
sino si (puntajeTot <= 13)
    bonificacion = 5.0*puntajeTot
sino si (puntajeTot <= 16)
    bonificacion = 7.5*puntajeTot
sino si (puntajeTot <= 19)
    bonificacion = 10.0*puntajeTot
sino
    bonificacion = 12.5*puntajeTot

// Salida de resultados
Imprimir puntajePun, puntajeRen, puntajeTot, bonificacion
Inicio

```

Programa



```

import java.awt.event.*;
import java.awt.*;
import javax.swing.*;

public class Programa5 extends JApplet implements ActionListener{
    // Declaración de variables
    JButton btnProcesar, btnBorrar;
    JTextArea txtS;
    JLabel lblMinutosTardanza, lblNumeroObservaciones;
    JTextField txtMinutosTardanza, txtNumeroObservaciones;
    JScrollPane scpScroll;

    // Crea la interfaz gráfica de usuario
    public void init() {
        getContentPane().setLayout(null);

        lblMinutosTardanza = new JLabel("Minutos de tardanza");
        lblMinutosTardanza.setBounds(15, 15, 160, 23);
        getContentPane().add(lblMinutosTardanza);

        lblNumeroObservaciones = new JLabel("Numero de observaciones");
        lblNumeroObservaciones.setBounds(15, 39, 160, 23);
        getContentPane().add(lblNumeroObservaciones);

        txtMinutosTardanza = new JTextField();
    }
}

```



```
txtMinutosTardanza.setBounds(175, 15, 120, 23);
getContentPane().add(txtMinutosTardanza);

txtNumeroObservaciones = new JTextField();
txtNumeroObservaciones.setBounds(175, 39, 120, 23);
getContentPane().add(txtNumeroObservaciones);

btnProcesar = new JButton("Procesar");
btnProcesar.setBounds(365, 15, 100, 23);
btnProcesar.addActionListener(this);
getContentPane().add(btnProcesar);

btnBorrar = new JButton("Borrar");
btnBorrar.setBounds(365, 39, 100, 23);
btnBorrar.addActionListener(this);
getContentPane().add(btnBorrar);

txtS = new JTextArea();
txtS.setFont(new Font("monospaced", 0, 12));

scpScroll = new JScrollPane(txtS);
scpScroll.setBounds(15, 77, 450, 100);
getContentPane().add(scpScroll);
}

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

        // Declaración de variables
        int minutosTar, numeroObs, puntajePun, puntajeRen, puntajeTot;
        double bonificacion;

        // Entrada de datos
        minutosTar = Integer.parseInt(txtMinutosTardanza.getText());
        numeroObs = Integer.parseInt(txtNumeroObservaciones.getText());

        // Determinación del puntaje por puntualidad
        if(minutosTar == 0)
            puntajePun = 10;
        else if(minutosTar <= 2)
            puntajePun = 8;
        else if(minutosTar <= 5)
            puntajePun = 6;
        else if(minutosTar <= 9)
            puntajePun = 4;
        else
            puntajePun = 0;

        // Determinación del puntaje por rendimiento
        if(numeroObs == 0)
            puntajeRen = 10;
        else if(numeroObs == 1)
            puntajeRen = 8;
        else if(numeroObs == 2)
            puntajeRen = 5;
        else if(numeroObs == 3)
            puntajeRen = 1;
        else
            puntajeRen = 0;

        // Determinación del puntaje total
        puntajeTot = puntajePun + puntajeRen;

        // Determinación de la bonificación
        if(puntajeTot < 11)
            bonificacion = 2.5*puntajeTot;
        else if(puntajeTot <= 13)
```

```
        bonificacion = 5.0*puntajeTot;
    else if(puntajeTot <= 16)
        bonificacion = 7.5*puntajeTot;
    else if(puntajeTot <= 19)
        bonificacion = 10.0*puntajeTot;
    else
        bonificacion = 12.5*puntajeTot;

    // Salida de resultados
    txtS.setText("Puntaje por puntualidad : " + puntajePun + "\n");
    txtS.append("Puntaje por rendimiento : " + puntajeRen + "\n");
    txtS.append("Puntaje total : " + puntajeTot + "\n");
    txtS.append("Bonificación : " + bonificacion);
}

if( e.getSource() == btnBorrar ){
    txtMinutosTardanza.setText("");
    txtNumeroObservaciones.setText("");
    txtS.setText("");
    txtMinutosTardanza.requestFocus();
}
}
```

Problemas propuestos

Actividad

1. Diseñe un programa que determine la categoría de un estudiante en base a su promedio ponderado de acuerdo a la siguiente tabla:

Promedio	Categoría
≥ 17	A
≥ 14 pero < 17	B
≥ 12 pero < 14	C
< 12	D

2. Una tienda vende tres tipos de productos cuyos códigos son 101, 102 y 103 a los precios unitarios dados en la siguiente tabla:

Código	Precio unitario
101	S/. 21.5
102	S/. 30.0
103	S/. 15.5

Como oferta la tienda ofrece un porcentaje de descuento sobre el importe de la compra de acuerdo a la siguiente tabla:

Importe Compra	Descuento
≥ 700	16%
≥ 500 pero < 700	14%
≥ 200 pero < 500	12%
< 200	10%

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

3. El índice de masa corporal (IMC) permite medir el grado de sobrepeso u obesidad de una persona. El IMC de una persona se calcula con la fórmula:

$$IMC = \frac{\text{peso}}{\text{estatura}^2}$$

estando el peso en kilogramos y la estatura en metros. En base al valor del IMC, se obtiene el grado de obesidad de la persona de acuerdo a la tabla adjunta. Diseñe un algoritmo que determine el grado de obesidad de una persona conociendo su peso y su estatura.

IMC	Grado de obesidad
< 20	Delgado
≥ 20 pero < 25	Normal
≥ 25 pero < 27	Sobrepeso
≥ 27	Obesidad

Autoevaluación

1. Diseñe un programa que lea un número entero en el intervalo de 1 a 4, correspondiente al estado civil de una persona y determine el nombre del estado civil. Considere: 1 para soltero, 2 para casado, 3 para viudo y 4 para divorciado. Si el número no está en el intervalo de 1 a 4, imprima un mensaje de error.
2. Diseñar un programa que lea la temperatura promedio de un día e imprima el tipo de clima correspondiente de acuerdo a la siguiente tabla:

Temperatura	Clima
≤ 10	Frío
>10 pero ≤ 20	Nublado
>20 pero ≤ 30	Caluroso
> 30	Trópico

3. Se denomina velocidad de escape a la velocidad mínima inicial que necesita un objeto para escapar de la gravitación de un cuerpo astronómico y continuar desplazándose sin tener que hacer otro esfuerzo propulsor. En la tabla adjunta se dan velocidades de escape para los planetas del sistema solar, en km/seg. Diseñe un programa que determine la velocidad de escape para un planeta del sistema solar.

Planeta	Velocidad de escape
Mercurio	4.2
Venus	10.3
Tierra	11.2
Marte	5.0
Júpiter	61.0
Saturno	36.0
Urano	22.0
Neptuno	24.0
Plutón	5.3
Luna	2.4

4. Dadas las estaturas de Juan, Pedro y Miguel, diseñe un programa que determine quienes de ellos tienen una estatura menor que la estatura promedio e imprima un mensaje como: "Juan y Pedro miden menos que el promedio" o "Pedro y Miguel miden menos que el promedio" o "Juan mide menos que el promedio", etc.

5. Los minerales de hierro se clasifican de acuerdo con su contenido de fósforo en la forma indicada en la tabla adjunta. Diseñe un programa que permita clasificar un mineral de hierro.

Porcentaje de fósforo	Clasificación
< 0.05	Bessemer
≥ 0.05 pero ≤ 0.18	No Bessemer
> 0.18	Fosforoso

6. Por la altura de sus tallos aéreos, las plantas se clasifican en la forma indicada en la tabla adjunta. Diseñe un programa que determine la denominación de un tallo.

Altura del tallo	Denominación
Hasta 1.00 m.	Mata
Más de 1.00 m pero no más que 4.00 m.	Arbusto
Más de 4.00 m pero no más que 8.00 m.	Arbolillo
Más de 8.00 m.	Árbol

7. Una compañía cobra las cuotas mensuales de sus clientes de acuerdo a lo siguiente:

- Si el cliente paga dentro de los primeros diez días del mes, obtiene un descuento igual al mayor valor entre \$5 y el 2% de la cuota.
- Si el cliente paga en los siguientes diez días, no tiene derecho a ningún descuento; deberá pagar exactamente la suma adeudada.
- Si el cliente paga dentro de los restantes días del mes, tendrá un recargo igual al mayor valor entre \$10 y el 3% de la cuota.

Diseñe un programa que determine cuanto debe pagar un cliente en un mes dado.

8. Una librería estima los precios de sus libros de la siguiente forma: el precio básico de un libro es de \$5.00 más \$0.15 por página. Sin embargo, si el número de páginas excede de 300, el precio sufrirá un recargo adicional de \$10. Además, si el número de páginas excede de 550, el precio se incrementará en otros \$7.50. Diseñe un programa que determine el precio de un libro.
9. Una empresa calcula el sueldo bruto de sus trabajadores multiplicando las horas trabajadas por una tarifa horaria que depende de la categoría del trabajador de acuerdo a la siguiente tabla:

Categoría	Tarifa
A	S/. 21.0
B	S/. 19.5
C	S/. 17.0
D	S/. 15.5

Por ley, todo trabajador se somete a un porcentaje de descuento del sueldo bruto: 20% si el sueldo bruto es mayor que S/. 2500 y 15% en caso contrario.

Diseñe un programa que determine el sueldo bruto, el descuento y el sueldo neto que le corresponden a un trabajador de la empresa.

10. Una empresa química paga a sus vendedores un sueldo bruto que es igual a la suma de un sueldo básico quincenal de S/.250 más una comisión igual a un porcentaje del total de las ventas efectuadas de acuerdo a la siguiente tabla:

Monto vendido	Comisión
≥ 20000	16%
≥ 15000 pero < 20000	14%
≥ 10000 pero < 15000	12%
< 10000	10%

Por otro lado, si el sueldo bruto del vendedor supera los S/.1800, este se somete a un descuento del 11%. Diseñe un programa que determine el sueldo bruto, el descuento y el sueldo neto de un vendedor.

11. Un curso se evalúa de la siguiente forma: se toman cinco prácticas calificadas, se determina el promedio de las cuatro notas más altas y se le da al estudiante una categoría que puede ser A, B, C ó D según la tabla siguiente:

Promedio	Categoría
≥ 17	A
≥ 14 pero < 17	B
≥ 10 pero < 14	C
< 10	D

Diseñe un programa que determine el promedio y la categoría de un estudiante.

12. En una empresa cada empleado tiene un código entero de tres cifras. Diseñe un programa que lea el código de un empleado y determine de que tipo de empleado se trata de acuerdo a los siguientes criterios:
- Si el código es divisible por 2, por 3 y por 5, el tipo de empleado es "Administrativo".
 - Si el código es divisible por 3 y por 5 pero no por 2, el tipo de empleado es "Directivo".
 - Si el código es divisible por 2, pero no por 3 ni por 5, el tipo de empleado es "Vendedor".
 - Si el código no es divisible por 2, ni por 3 ni por 5, el tipo de empleado es "Seguridad".
13. En una librería obsequian lapiceros Lucas, Cross y/o Novo por la compra de cuadernos de acuerdo a lo siguiente:

- Si el número de cuadernos adquiridos es menos que 12, no se obsequia ningún lapicero.
- Si el número de cuadernos adquiridos es no menos que 12, pero menos que 24, se obsequia 1 lapicero Lucas por cada 4 cuadernos adquiridos.
- Si el número de cuadernos adquiridos es no menos que 24, pero menos que 36, se obsequia 2 lapiceros Cross por cada 4 cuadernos adquiridos.
- Si el número de cuadernos adquiridos es no menos que 36, se obsequia 2 lapiceros Novo por cada 4 cuadernos adquiridos, más 1 lapicero Lucas y más 1 lapicero Cross.

Diseñe un programa que determine cuantos lapiceros Lucas, Cross y Novo recibe un cliente como obsequio.

14. Diseñe un programa que lea tres números a, b, c y determine si los números fueron ingresados en orden ascendente, en orden descendente o en desorden.
15. Diseñe un programa que lea un número entero y determine si tiene 1, 2, 3, 4 ó más de 4 dígitos.
16. Una empresa registra el sexo, edad y estado civil de sus empleados a través de un número entero positivo de cuatro cifras de acuerdo a lo siguiente: la primera cifra de la izquierda representa el estado civil (1 para soltero, 2 para casado, 3 para viudo y 4 para divorciado), las siguientes dos cifras representan la edad y la tercera cifra representa el sexo (1 para femenino y 2 para masculino). Diseñe un programa que determine el estado civil, edad y sexo de un empleado conociendo el número que empaqueta dicha información.
17. Se desea un programa para obtener el grado de eficiencia de un operario de torno de una fábrica productora de tornillos de acuerdo a las siguientes condiciones que se le impone para un período de 15 días.

Condiciones impuestas al operario:

- a. No más de 1.5 horas de ausencia al trabajo.
- b. Menos de 300 tornillos defectuosos producidos.
- c. Más de 10000 tornillos no defectuosos producidos.

Los grados de eficiencia para cada trabajador son asignados de la siguiente manera:

- Si no cumple ninguna condición, grado 5
- Si sólo cumple la primera condición, grado 7
- Si sólo cumple la segunda condición, grado 8
- Si sólo cumple la tercera condición, grado 9
- Si cumple la primera y segunda condición, grado 12
- Si cumple la primera y tercera condición, grado 13
- Si cumple la segunda y tercera condición, grado 15
- Si cumple las tres condiciones, grado 20

18. Una compañía alquila automóviles a los costos por día dados en la siguiente tabla:

Días de alquiler	Costo por día
1	\$ 50
2	\$ 45
3 a 5	\$ 40
6 a 10	\$ 35
Más de 10	\$ 30

Adicionalmente, si el cliente alquila un automóvil por más de 10 días, la compañía le obsequia una agenda; en caso contrario, sólo le obsequia un cuaderno.

Diseñe un programa que determine el importe que debe pagar un cliente por el alquiler de un automóvil y el obsequio que le corresponde.

19. Planos S.A se dedica al fotocopiado de planos topográficos a los precios indicados en la siguiente tabla:

Plano	Costo por copia
Menos de 500 hectáreas	S/. 25
De 501 a 1000 hectáreas	S/. 30
De 1001 a 2000 hectáreas	S/. 40
De 2001 a 5000 hectáreas	S/. 50
De 5001 a 10000 hectáreas	S/. 75
Más de 10000 hectáreas	S/. 100

Como oferta, Planos S.A ofrece un descuento del 5% para la segunda copia y del 10% para cada una de las demás copias.

Diseñe un programa que determine el importe bruto, el importe de descuento (0 si no hay descuento) y el importe a pagar por cierto número de fotocopias de un plano.

20. Un vivero municipal vende plántones a los precios indicados en la siguiente tabla:

Plantones	Costo por unidad
Forestal	S/. 0.35
Forestal ornamental	S/. 0.50
Frutal	S/. 2.00
Frutal injertado	S/. 3.00
Bonsái	S/. 5.00

Por otro lado, si el cliente adquiere más de 10 docenas de plántones de cualquier tipo, el vivero le obsequia 3 plántones por cada docena adquirida; en caso contrario, sólo le obsequia 1 plántón por cada docena.

Diseñe un programa que determine el importe a pagar y el número de plántones de obsequio por la compra de cierta cantidad de plántones de un mismo tipo.

21. En una elección democrática a la presidencia de un club femenino participan Débora, Raquel y Séfora. Para ganar la elección se requiere obtener la mitad de los votos emitidos más uno. En caso de no haber un ganador, pasan a una segunda vuelta los candidatos que alcanzaron los dos primeros puestos o se

anula la elección, si hay empate entre los tres o, si hay empate por el segundo puesto. Dados los votos obtenidos por cada candidato, se le pide diseñar un programa que determine el nombre del candidato ganador o los nombres de los candidatos que pasan a la segunda vuelta o un mensaje indicando la anulación de la elección.

22. Escriba una versión más simple para cada uno de los siguientes fragmentos de código:

```
a.  if(a > 2 || b > 2 || c > 2)
    x = 1;
    else if(a%2 == 1 && b <= 2)
    x = 2;
    else if(b%2 == 1 && c <= 2)
    x = 3;
    else if(c%2 == 1 && a <= 2)
    x = 4;
    else
    x = 5;
```

```
b.  if(x%3 == 0){
    a = 0;
    b = 2;
    c = 3;
}
    else if(x%2 == 1){
    a = 1;
    b = 2;
    c = 3;
}
    else{
    a = 2;
    b = 2;
    c = 3;
}
```

23. Explique y corrija los errores de sintaxis de los siguientes fragmentos de programa.

```
a.  if(a > 20)
    x = 1;
    y = 2;
    else if(a > 15)
    x = 2;
    y = 3;
    else{
    x = 3;
    y = 4;
}
```

```
b.  if(x%3 == 0){
    a = 0;
    else if(x%2 == 1)
    a = 1;
    else
    a = 2;
}
```

24. Determine que imprime el siguiente fragmento de programa para los siguientes casos: a). x igual a 1, b). x igual a 2, c). x igual a 7 y d). x igual a 5

```
if(x == 1){
    txtS.append("aaaa\n");
    x = x + 1;
}
if(x == 2){
    txtS.append("bbbb\n");
    x = x + 2;
}
if(x >= 8)
    txtS.append("cccc\n");
else if(x >= 6)
    txtS.append("dddd\n");
else
    txtS.append("eeee\n");
txtS.append("ffff");
```

Para recordar

- Colocar ; al final de cualquier condición de la estructura **if...else...if** causa un error de sintaxis.
- En caso que las acciones de un **if**, de un **else if** o de un **else** sean compuestas (varias acciones simples), estas deben ir entre llaves de bloque { }.

**UNIDAD DE
APRENDIZAJE****3****SEMANA****9**

ESTRUCTURA DE SELECCIÓN MÚLTIPLE SWITCH

LOGRO DE LA UNIDAD DE APRENDIZAJE

Al finalizar la unidad, los alumnos, utilizando estructuras algorítmicas de selección, diseñarán algoritmos en pseudocódigo que resuelvan problemas que involucren tomas de decisiones.

TEMARIO

1. La estructura de selección Múltiple switch.

ACTIVIDADES

Los alumnos desarrollan algoritmos que involucren estructuras de selección múltiple.

1. LA ESTRUCTURA DE SELECCIÓN MÚLTIPLE **switch**

La *estructura de selección múltiple switch* permite seleccionar una ruta de entre varias rutas posibles en base al valor de una variable **selector** que se compara con una lista de constantes enteras o de carácter **c1**, **c2**, **c3**, ..., **cn**. Cuando se encuentra una correspondencia entre el valor de la variable **selector** y una constante, se ejecuta la acción o el grupo de acciones asociadas a dicha constante. Si el selector no coincide con ninguna constante, se efectúa la acción por defecto, si es que existe.

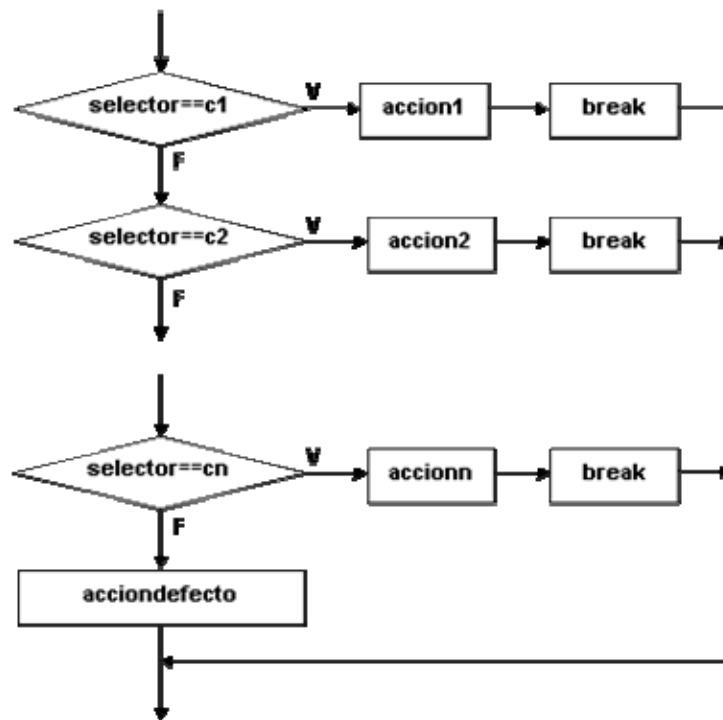


Figura 6.1 Estructura de Selección Múltiple **switch**

En la tabla que sigue se muestra el código y el pseudocódigo de la estructura **switch**.

Código Java	Pseudocódigo
<pre> switch(selector){ case c1: accion1; break; case c2: accion2; break; . . . case cn: accionnn; break; default : acciondefecto; } </pre>	<pre> segun(selector){ caso c1: accion1 caso c2: accion2 . . . caso cn: accionnn defecto: acciondefecto } </pre>

Consideraciones:

- Las sentencias **break** y el caso por defecto **default** son opcionales.
- El caso por defecto **default** no tiene que ser el último de todos sino que puede ser el primero u ocupar una posición intermedia.
- Luego de efectuarse la acción o las acciones de un **case** o del **default**, se proseguirá con la ejecución de la acción o las acciones de los case que siguen hasta encontrar un **break** o hasta llegar al final de la estructura **switch**; lo que ocurra primero.
- Es un error de sintaxis tener casos duplicados.
- Las acciones pueden ser acciones simples o acciones compuestas. En el caso de acciones compuestas no es necesario colocarlas entre llaves de bloque.

2. PROBLEMAS RESUELTOS**Problema 6. 1**

Diseñe un programa que determine la cantidad de días y el nombre de un mes conociendo los valores numéricos del mes y del año.

Algoritmo

```
Inicio
  // Declaración de variables
  entero dias, mes, año
  cadena nombre

  // Entrada de datos
  Leer mes, año

  // Determina el nombre del mes
  según( mes ){
    caso 1 : nombre = "Enero"
    caso 2 : nombre = "Febrero"
    caso 3 : nombre = "Marzo"
    caso 4 : nombre = "Abril"
    caso 5 : nombre = "Mayo"
    caso 6 : nombre = "Junio"
    caso 7 : nombre = "Julio"
    caso 8 : nombre = "Agosto"
    caso 9 : nombre = "Setiembre"
    caso 10: nombre = "Octubre"
    caso 11: nombre = "Noviembre"
    defecto: nombre = "Diciembre"
  }

  // Determina el número de días del mes
  según( mes ){
    caso 1:
    caso 3:
    caso 5:
    caso 7:
    caso 8:
    caso 10:
    caso 12:
      dias = 31
    caso 4:
    caso 6:
    caso 9:
    caso 11:
      dias = 30
    defecto:
```

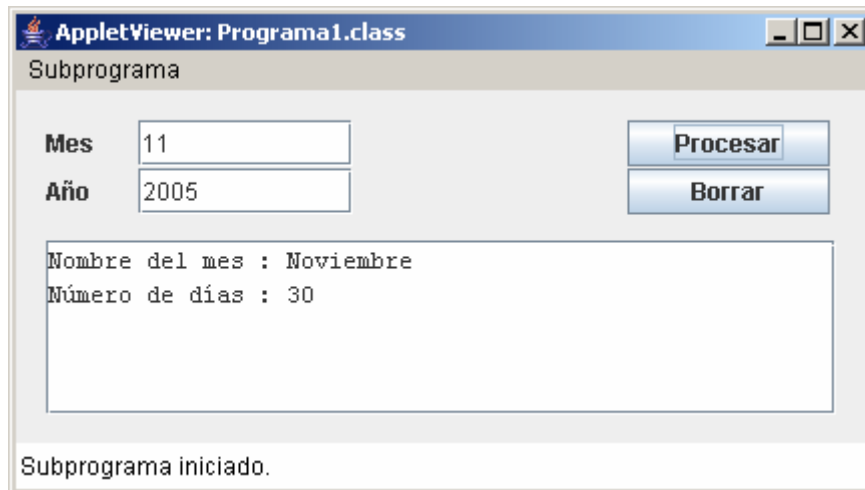
```

        si((año%4 == 0) && ((año%100 != 0) || (año%400 == 0)))
            dias = 29
        sino
            dias = 28
    }

    // Salida
    Imprimir nombre, dias
Fin

```

Programa



```

import java.awt.event.*;
import java.awt.*;
import javax.swing.*;

public class Programa1 extends JApplet implements ActionListener{
    // Declaración de variables
    JButton btnProcesar, btnBorrar;
    JTextArea txtS;
    JLabel lblAño, lblMes;
    JTextField txtAño, txtMes;
    JScrollPane scpScroll;

    // Crea la interfaz gráfica de usuario GUI
    public void init() {
        getContentPane().setLayout(null);

        txtMes = new JTextField();
        txtMes.setBounds(61, 15, 107, 23);
        getContentPane().add(txtMes);

        txtAño = new JTextField();
        txtAño.setBounds(61, 39, 107, 23);
        getContentPane().add(txtAño);

        btnProcesar = new JButton("Procesar");
        btnProcesar.setBounds(305, 15, 101, 23);
        btnProcesar.addActionListener(this);
        getContentPane().add(btnProcesar);

        btnBorrar = new JButton("Borrar");
        btnBorrar.setBounds(305, 39, 101, 23);
        btnBorrar.addActionListener(this);
        getContentPane().add(btnBorrar);

        lblAño = new JLabel("Año");
        lblAño.setBounds(15, 39, 42, 23);
    }
}

```



```
getContentPane().add(lblAño);

lblMes = new JLabel("Mes");
lblMes.setBounds(15, 15, 42, 23);
getContentPane().add(lblMes);

txtS = new JTextArea();
txtS.setFont(new Font("monospaced", 0, 12));

scpScroll = new JScrollPane(txtS);
scpScroll.setBounds(15, 75, 394, 87);
getContentPane().add(scpScroll);
}

// Procesa eventos de tipo ActionEvent
public void actionPerformed( ActionEvent e ){

    // Si se hizo clic en Procesar
    if( e.getSource() == btnProcesar ){

        // Declaración de variables
        int dias, mes, año;
        String nombre;

        // Entrada de datos
        mes = Integer.parseInt(txtMes.getText());
        año = Integer.parseInt(txtAño.getText());

        // Determina el nombre del mes
        switch( mes ){
            case 1 :
                nombre = "Enero";
                break;
            case 2 :
                nombre = "Febrero";
                break;
            case 3 :
                nombre = "Marzo";
                break;
            case 4 :
                nombre = "Abril";
                break;
            case 5 :
                nombre = "Mayo";
                break;
            case 6 :
                nombre = "Junio";
                break;
            case 7 :
                nombre = "Julio";
                break;
            case 8 :
                nombre = "Agosto";
                break;
            case 9 :
                nombre = "Setiembre";
                break;
            case 10:
                nombre = "Octubre";
                break;
            case 11:
                nombre = "Noviembre";
                break;
            default:
                nombre = "Diciembre";
        }

        // Determina el número de días del mes
```

```

switch( mes ){
    case 1: case 3: case 5: case 7: case 8: case 10: case 12:
        dias = 31;
        break;
    case 4: case 6: case 9: case 11:
        dias = 30;
        break;
    default:
        if((año%4 == 0) && ((año%100 != 0) || (año%400 == 0)))
            dias = 29;
        else
            dias = 28;
}

// Salida de resultados
txtS.setText("Nombre del mes : " + nombre + "\n");
txtS.append("Número de días : " + dias);
}

// Si se hizo clic en Borrar
if( e.getSource() == btnBorrar ){
    txtS.setText("");
    txtMes.setText("");
    txtAño.setText("");
    txtMes.requestFocus();
}
}
}

```

Problema 6.2

Dados los siguientes tipos de papel y sus respectivas dimensiones:

Tamaño de papel	Dimensiones
A4	297 x 210 mm
B5	182 x 257 mm
A5	148 x 210 mm
Carta	8½ x 11 pulg
Legal	8½ x 14 pulg
Ejecutivo	7¼ x 10½ pulg
Media carta	5½ x 8½ pulg

Se conoce como área imprimible al área que queda libre luego de descontar los márgenes superior, inferior, izquierdo y derecho. Diseñe un programa que permita obtener el área imprimible de un tamaño de papel elegido, en cm².

Algoritmo

```

Inicio
    // Declaración de variables
    entero tamaño
    real mrgsup, mrginf, mrgder, mrgizq, ancho, alto, area

    // Entrada de datos
    Leer tamaño, mrgsup, mrginf, mrgder, mrgizq

    // Determina el ancho y el alto del papel en cm
    según(tamaño){
        caso 0:
            ancho = 29.7
            alto = 21.0

```

```

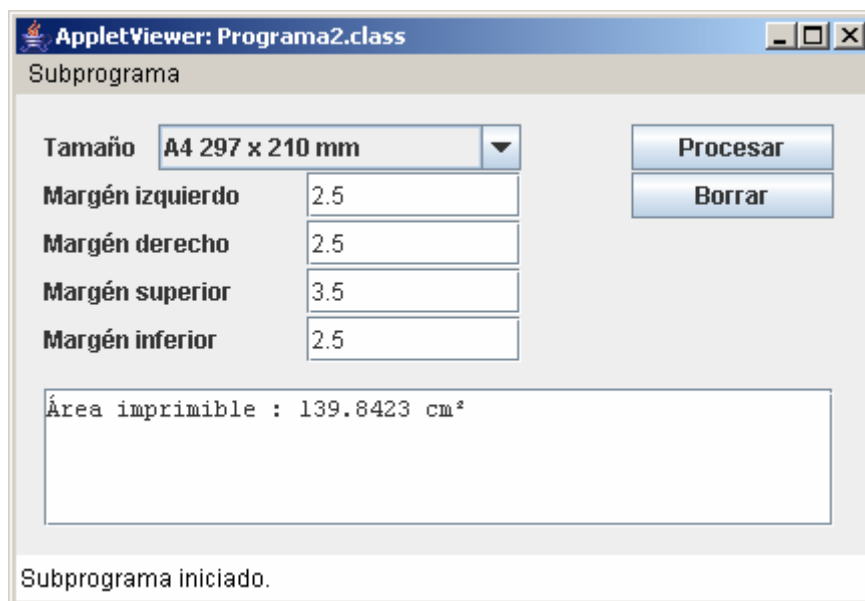
    caso 1:
        ancho = 18.2
        alto = 25.7
    caso 2:
        ancho = 14.8
        alto = 21.0
    caso 3:
        ancho = 8.5*2.54
        alto = 11*2.54
    caso 4:
        ancho = 8.5*2.54
        alto = 14*2.54
    caso 5:
        ancho = 7.25*2.54
        alto = 10.5*2.54
    defecto:
        ancho = 5.5*2.54
        alto = 8.5*2.54
}

// Calcula el area imprimible
area = (ancho-mrgizq-mrgder)*(alto-mrgsup-mrginf)

// Muestra al área imprimible
Imprimir area
Fin

```

Programa



```

import java.awt.event.*;
import java.awt.*;
import javax.swing.*;

public class Programa2 extends JApplet implements ActionListener{
    // Declaración de variables
    JButton btnProcesar, btnBorrar;
    JTextArea txtS;
    JComboBox cboTamaño;
    JLabel lblMrgDer, lblMrgInf, lblMrgIzq, lblMrgSup, lblTamaño;
    JTextField txtMrgDer, txtMrgInf, txtMrgIzq, txtMrgSup;
    JScrollPane scpScroll;

    // Crea la interfaz gráfica de usuario GUI

```

```

public void init() {
    getContentPane().setLayout(null);
    setBackground(new Color(214,211,206));

    txtMrgIzq = new JTextField();
    txtMrgIzq.setBounds(145, 39, 107, 23);
    getContentPane().add(txtMrgIzq);

    txtMrgDer = new JTextField();
    txtMrgDer.setBounds(145, 63, 107, 23);
    getContentPane().add(txtMrgDer);

    txtMrgSup = new JTextField();
    txtMrgSup.setBounds(145, 87, 107, 23);
    getContentPane().add(txtMrgSup);

    txtMrgInf = new JTextField();
    txtMrgInf.setBounds(145, 111, 107, 23);
    getContentPane().add(txtMrgInf);

    btnProcesar = new JButton("Procesar");
    btnProcesar.setBounds(307, 15, 101, 23);
    btnProcesar.addActionListener(this);
    getContentPane().add(btnProcesar);

    btnBorrar = new JButton("Borrar");
    btnBorrar.setBounds(307, 39, 101, 23);
    btnBorrar.addActionListener(this);
    getContentPane().add(btnBorrar);

    cboTamaño = new JComboBox();
    cboTamaño.setBounds(71, 15, 181, 23);
    cboTamaño.addItem("A4 297 x 210 mm");
    cboTamaño.addItem("B5 182 x 257 mm");
    cboTamaño.addItem("A5 148 x 210 mm");
    cboTamaño.addItem("Carta 8½ x 11 pulg");
    cboTamaño.addItem("Legal 8½ x 14 pulg");
    cboTamaño.addItem("Ejecutivo 7¼ x 10½ pulg");
    cboTamaño.addItem("Media carta 5½ x 8½ pulg");
    getContentPane().add(cboTamaño);

    lblMrgDer = new JLabel("Margén derecho");
    lblMrgDer.setBounds(14, 63, 124, 23);
    getContentPane().add(lblMrgDer);

    lblMrgInf = new JLabel("Margén inferior");
    lblMrgInf.setBounds(14, 111, 124, 23);
    getContentPane().add(lblMrgInf);

    lblMrgIzq = new JLabel("Margén izquierdo");
    lblMrgIzq.setBounds(14, 39, 124, 23);
    getContentPane().add(lblMrgIzq);

    lblMrgSup = new JLabel("Margén superior");
    lblMrgSup.setBounds(14, 87, 124, 23);
    getContentPane().add(lblMrgSup);

    lblTamaño = new JLabel("Tamaño");
    lblTamaño.setBounds(14, 15, 49, 23);
    getContentPane().add(lblTamaño);

    txtS = new JTextArea();
    txtS.setFont(new Font("monospaced", 0, 12));

    scpScroll = new JScrollPane(txtS);
    scpScroll.setBounds(14, 147, 394, 69);
    getContentPane().add(scpScroll);
}

```

```

// Procesa eventos de tipo(ActionEvent)
public void actionPerformed( ActionEvent e ){

    // Si se hizo clic en Procesar
    if( e.getSource() == btnProcesar ){

        // Declaración de variables
        int tamaño;
        double mrgsup, mrginf, mrgder, mrgizq, ancho, alto, area;

        // Entrada de datos
        tamaño = cboTamaño.getSelectedIndex();
        mrgsup = Double.parseDouble(txtMrgSup.getText());
        mrginf = Double.parseDouble(txtMrgInf.getText());
        mrgder = Double.parseDouble(txtMrgDer.getText());
        mrgizq = Double.parseDouble(txtMrgIzq.getText());

        // Determina el ancho y el alto del papel en cm
        switch(tamaño){
            case 0:
                ancho = 29.7;
                alto = 21.0;
                break;
            case 1:
                ancho = 18.2;
                alto = 25.7;
                break;
            case 2:
                ancho = 14.8;
                alto = 21.0;
                break;
            case 3:
                ancho = 8.5*2.54;
                alto = 11*2.54;
                break;
            case 4:
                ancho = 8.5*2.54;
                alto = 14*2.54;
                break;
            case 5:
                ancho = 7.25*2.54;
                alto = 10.5*2.54;
                break;
            default:
                ancho = 5.5*2.54;
                alto = 8.5*2.54;
        }

        // Calcula el area imprimible
        area = (ancho-mrgizq-mrgder)*(alto-mrgsup-mrginf);

        // Muestra al área imprimible
        txtS.setText("Área imprimible : " + area + "cm²");
    }

    // Si se hizo clic en Borrar
    if( e.getSource() == btnBorrar ){
        txtMrgIzq.setText("");
        txtMrgDer.setText("");
        txtMrgSup.setText("");
        txtMrgInf.setText("");
        txtS.setText("");
        txtMrgIzq.requestFocus();
    }
}
}

```

Problema 6.3

En el siguiente ejercicio, reemplaze la estructura **if...else...if** por la estructura **switch**. Considere que **producto** es de tipo **int**.

```
if( producto == 0 )
    precio = 25;
else if( producto == 1 )
    precio = 15;
else if( producto == 2 )
    precio = 10;
else
    precio = 12;
```

Solución

El equivalente **switch** es el siguiente:

```
switch( producto ){
    case 0:
        precio = 25;
        break;
    case 1:
        precio = 15;
        break;
    case 2:
        precio = 10;
        break;
    default:
        precio = 12;
}
```

Problema 6. 4

En el siguiente ejercicio, reemplaze la estructura **if...else...if** por la estructura **switch**. Considere que **z** es de tipo **int**.

```
if( z == 0 )
    a = 10;
else if( z == 1 || z == 3 || z == 5 )
    a = 2;
else if( z == 2 || z == 4 )
    a = 7;
else
    a = 3;
```

El equivalente **switch** es el siguiente:

```
switch( z ){
    case 0:
        a = 10;
        break;
    case 1:
    case 3:
    case 5:
        a = 2;
        break;
    case 2:
    case 4:
        a = 7;
        break;
    default:
        a = 3;
```

}

Problema 6.5

Diseñe un algoritmo que determine la cifra de las unidades de un número natural como palabra.

Algoritmo**Inicio**

```
// Declaración de variables
entero numero, unidades
cadena palabra

// Entrada de datos
Leer numero

// Determine la cifra de las unidades como número
unidades = numero%10

// Determine la cifra de las unidades como palabra
segun( unidades ){
    caso 0 : palabra = "cero"
    caso 1 : palabra = "uno"
    caso 2 : palabra = "dos"
    caso 3 : palabra = "tres"
    caso 4 : palabra = "cuatro"
    caso 5 : palabra = "cinco"
    caso 6 : palabra = "seis"
    caso 7 : palabra = "siete"
    caso 8 : palabra = "ocho"
    defecto: palabra = "nueve"
}

// Salida de resultados
Imprimir palabra
```

Fin

Problemas propuestos

Actividad

1. Una heladería vende helados a los precios unitarios dados en la siguiente tabla:

Helado	Precio unitario
Sol	S/. 1.5
Fresa	S/. 2.0
Mar	S/. 1.7
Rico	S/. 2.5

Diseñe un algoritmo que determine el importe a pagar por la compra de cierta cantidad de helados del mismo tipo.

2. Reemplace la estructura **if...else...if** por la estructura **switch**. Considere que **n** es de tipo **int**.

```

if( n == 0 )
    z = 1;
else if( n == 7 || n == 4 ){
    z = 2;
    if( a < b)
        z = 7;
}
else if( n == 5 )
    z = 5;
else
    z = 8;

```

3. Diseñe un algoritmo que califique el puntaje obtenido en el lanzamiento de un dado de acuerdo a la siguiente tabla:

Puntaje	Calificación
1 ó 2	Pésimo
3 ó 4	Regular
5	Muy bien
6	Excelente

4. La empresa *Construya Fácil* ha puesto a la venta los libros indicados en la siguiente tabla:

Puntaje	Precio en la sede
Manual Práctico de Construcción	\$ 30
Manual Práctico de Instalaciones Sanitarias	\$ 27
Manual Práctico de Instalaciones Eléctricas	\$ 20
Manual Práctico de Acabados	\$ 35

Los precios indicados en la tabla se aplican en caso que el pago se efectúe en la sede de la empresa. Si el pago se efectúa a domicilio, el precio se incrementa en un 2%. Dada una compra de cierta cantidad de unidades de un mismo título de

libro, diseñe un algoritmo que determine el importe total a pagar y el precio unitario pagado por cada libro.

Autoevaluación

1. Diseñe un programa que lea un número entero en el intervalo de 1 a 7, correspondiente a un día de la semana y determine el nombre del día. Considere: 1 para lunes, 2 para martes, 3 para miércoles, 4 para jueves, 5 para viernes, 6 para sábado y 7 para domingo. Si el número no está en el intervalo de 1 a 7, imprima un mensaje de error.
2. En un instituto, los cursos de matemática, física y química se evalúan en base a tres prácticas calificadas. Cada práctica tiene un peso dado en la siguiente tabla:

Curso	Práctica 1	Práctica 2	Práctica 3
Matemática	1	2	3
Física	1	1	2
Química	1	1	1

Dadas las notas de práctica de un alumno para uno de los tres cursos, diseñe un programa que determine el peso de cada práctica, el promedio final del alumno y su condición de aprobado o desaprobado. La nota mínima aprobatoria es 13.

3. En un cine, los precios de las entradas son los mostrados en la siguiente tabla:

Día	General	Niños
Lunes	S/. 9	S/.7
Martes	S/. 7	S/.7
Miércoles a Viernes	S/. 10	S/.8
Sábado y Domingo	S/. 12	S/.9

Como oferta, el cine entrega un vale por media entrada si el número total de entradas adquiridas (la suma del número de entradas generales más el número de entradas de niños) es mayor que 5. Diseñe un programa que determine cuánto debe pagar un cliente por la compra de cierta cantidad de entradas generales y/o de niños.

4. Determine que imprime el siguiente fragmento de programa para cada uno de los siguientes casos: **a).** v igual a 1, **b).** v igual a 2, **c).** v igual a 3 y **d).** v igual a 4. Considere que la variable **v** es de tipo **int**.

```
switch(v) {
    case 1:
        txtS.append("Uno\n");
    case 2:
        txtS.append("Dos\n");
    case 3:
        txtS.append("Tres\n");
    default:
        txtS.append("Defecto\n");
}
```

5. Determine que imprime el siguiente fragmento de programa para v igual a 5. Considere que v es de tipo `int`.

```
switch(v) {  
    case 1:  
        txtS.append("Uno\n");  
    case 2:  
        txtS.append("Dos\n");  
    default:  
        txtS.append("Defecto\n");  
    case 3:  
        txtS.append("Tres\n");  
}
```

6. Reescriba los siguientes fragmentos de programa usando la estructura **if-else-if**.

a.

```
switch(estacion) {  
    case 0:  
        txtS.append("Primavera");  
        break;  
    case 1:  
        txtS.append("Verano");  
        break;  
    case 2:  
        txtS.append("Otoño");  
        break;  
    default:  
        txtS.append("Invierno")  
}
```

b.

```
switch(numero) {  
    case 0: case 1: case 2:  
        x = 10;  
        break;  
    case 3: case 4: case 5:  
        x = 15;  
        break;  
    default:  
        x = 20;  
}
```

Para recordar

- Si un **case** no tiene **break**, sucederá que al ejecutar las acciones de dicho **case** se ejecutarán, también, las acciones de los **case** que siguen hasta encontrar un **break** o hasta llegar al final del **switch**.
- Se puede usar la estructura **switch** en una toma de decisiones únicamente si las condiciones consisten en comparaciones de una misma variable con una lista de constantes enteras o de carácter.

UNIDAD DE
APRENDIZAJE

4

SEMANA

10

MÉTODOS TIPO VOID

LOGRO DE LA UNIDAD DE APRENDIZAJE

Al finalizar la unidad, los alumnos, utilizando descomposición en módulos mediante métodos tipo void y con valor de retorno y, los demás conceptos aprendidos en el curso, diseñarán programas en Java que resuelvan problemas planteados.

CONTENIDO

1. Programación modular.
2. Variables locales y globales.
3. Métodos tipo void.

ACTIVIDADES

Los alumnos desarrollan programas mediante descomposición modular.

1. PROGRAMACION MODULAR

La programación modular es una metodología de programación que permite construir un programa grande descomponiéndolo en pequeños subprogramas o módulos. Para ello se parte de un módulo principal que se descompone en varios submódulos que son controlados por el módulo principal. Si la tarea asignada a un módulo es demasiado compleja este deberá descomponerse en otros módulos más pequeños hasta lograr módulos que hagan tareas relativamente sencillas. A este proceso de refinamiento sucesivo se conoce también como la técnica de “divide y vencerás”.

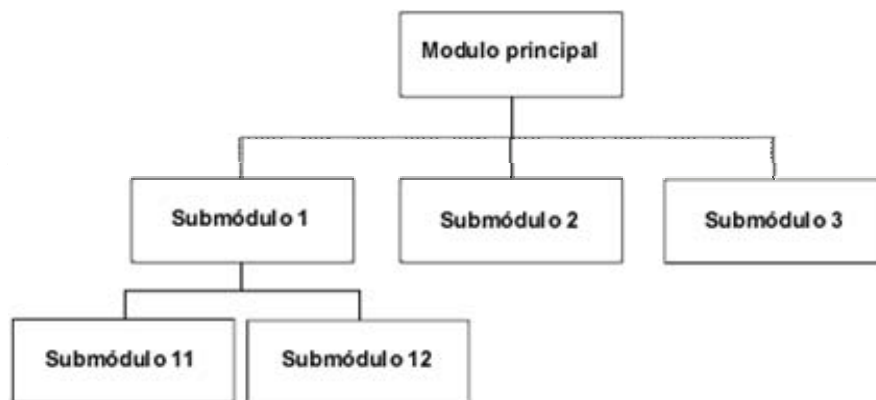


Figura 7.1 Descomposición modular de un programa

Las tareas asignadas a los subprogramas pueden ser de diversa índole: entrada, salida, cálculos, control de otros módulos, etc. Para que un subprograma pueda efectuar su tarea tiene que ser llamado o invocado por el programa principal o por algún otro módulo que considere necesario el servicio del subprograma. Una vez que el subprograma termina su tarea, devuelve el control al punto donde se hizo la llamada. Un subprograma puede llamar a su vez a otros subprogramas.

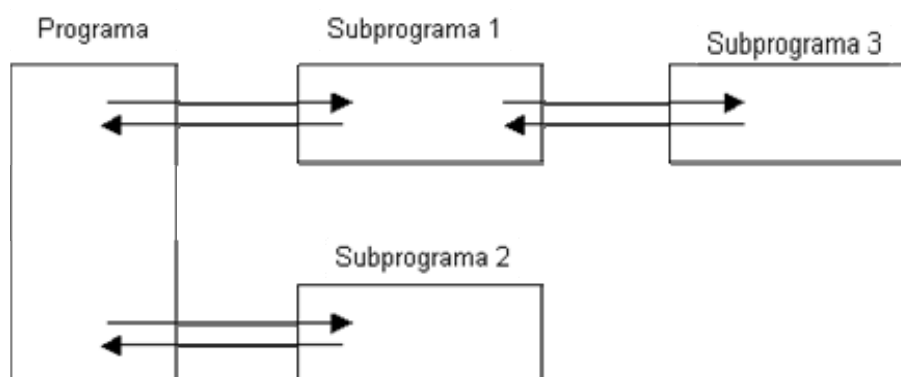


Figura 7.2 Un programa con diferentes niveles de subprograma.

En el lenguaje Java a los módulos o subprogramas se denominan **métodos**, mientras que en el lenguaje algorítmico se denominan **subalgoritmos**.

2. VARIABLES LOCALES Y GLOBALES

Los métodos pueden utilizar sus propias variables denominadas **variables locales** o variables de uso compartido, comunes a todos los métodos, denominadas **variables globales**.

2.1 Variables Locales

Una *variable local* es una variable que se declara en el interior de un método por lo que su ámbito es el interior del método, es decir, sólo puede ser utilizada dentro del método donde fue declarada. Este tipo de variable se crea al iniciar la ejecución del método y se destruye al finalizar. Por otro lado, una variable local se crea vacía, es decir no recibe ninguna inicialización automática.

2.2 Variables Globales

Una *variable global* es una variable que se declara dentro del programa, pero en el exterior de todos los métodos, por lo que su ámbito es el interior de todo el programa, es decir, puede ser utilizada desde cualquier parte del programa. Este tipo de variable se crea al iniciar la ejecución del programa y se destruye al finalizar. Por otro lado, una variable global se inicializa automáticamente: **0** si es de tipo **int**, **0.0** si es de tipo **double**, **false** si es de tipo **boolean**, **'\0'** si es de tipo **char** y **null** si es de tipo **String**.

3. METODOS TIPO void

Un *método tipo void* es un módulo de programa que puede recibir datos de entrada a través de variables locales denominadas parámetros; pero, que no retorna ningún resultado al punto donde es invocado, razón por el que se le conoce también como **método sin valor de retorno**. Este tipo de método, al igual que los métodos que retornan un valor, pueden recibir datos de entrada a través de variables locales al método conocidas como **parámetros**.

Los métodos tipo void pueden dividirse a su vez en dos tipos:

- Métodos tipo void sin parámetros.
- Métodos tipo void con parámetros.

3.1 Método tipo void sin parámetros

Estos métodos no pueden recibir datos de entrada ni retornar ningún resultado al punto de su invocación.

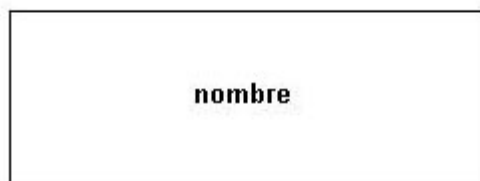


Figura 7.3 Método tipo void sin parámetros

Cuando se programa usando métodos se siguen dos etapas. Primero, el método debe definirse. Esto consiste en crear el método ubicándolo en alguna parte del programa. Segundo, el método creado debe ser invocado en el lugar donde se requiera. Esto consiste en poner el método en ejecución.

Definición

Este tipo de método se define de la siguiente manera:

```
void nombre() {
    Declaración de variables locales
    Cuerpo del método
}
```

Invocación

Este tipo de método se invoca de la siguiente manera:

```
nombre ();
```

Donde **nombre** es el nombre del método.

3.2 Métodos tipo void con parámetros

Estos métodos reciben datos de entrada a través de variables locales al método denominadas parámetros; pero, igual que en el caso anterior no pueden retornar ningún resultado al punto de su invocación.



Figura 7.4 Método tipo void con parámetros

Donde **p1**, **p2**, **p3**, etc son los parámetros del método. El número de parámetros es variable y depende de las necesidades del método.

Definición

Este tipo de método se define de la siguiente manera:

```
void nombre( tipo1 p1, tipo2 p2, tipo3 p3, . . . ){
    Declaración de variables locales
    Cuerpo del método
}
```

Donde:

nombre	: Es el nombre del método
p1, p2, p3, ...	: Son los nombres de los parámetros
tipo1, tipo2, tipo3, ...	: Son los tipos de datos de los parámetros

Llamada

Este tipo de método se invoca de la siguiente manera:

```
nombre (v1, v2, v3, . . . );
```

Donde:

nombre : Es el nombre del método invocado
v1, v2, v3, ... : Son los valores dados a los parámetros

Problema 7.1

En una universidad, los alumnos están categorizados en cuatro categorías. A cada categoría le corresponde una pensión mensual distinta dada en la siguiente tabla:

Categoría	Pensión
A	S/. 550
B	S/. 500
C	S/. 460
D	S/. 400

Semestralmente, la universidad efectúa rebajas en las pensiones de sus estudiantes a partir del segundo ciclo en base al promedio ponderado del ciclo anterior en porcentajes dados en la tabla siguiente:

Promedio	Descuento
0 a 13.99	No hay descuento
14.00 a 15.99	10 %
16.00 a 17.99	12 %
18.00 a 20.00	15 %

Diseñe un programa que determine cuánto de rebaja recibirá un estudiante sobre su pensión actual y a cuánto asciende su nueva pensión.

Declare todas las variables como globales y use métodos tipo void.

Programa

```
import java.awt.event.*;
```

```

import java.awt.*;
import javax.swing.*;

public class MetodosSVR1 extends JApplet implements ActionListener{
    // Declaración de variables globales para la GUI
    JButton btnProcesar, btnBorrar;
    JTextArea txtS;
    JComboBox choCategoria;
    JLabel lblCategoria, lblPromedio;
    JTextField txtPromedio;
    JScrollPane scpScroll;

    // Declaración de variables globales para el algoritmo
    int categoria;
    double actualpen, nuevapen, descuento, promedio;

    // -----
    // Crea la interfaz gráfica de usuario GUI
    public void init() {
        getContentPane().setLayout(null);
        setBackground(new Color(214,211,206));

        txtPromedio = new JTextField();
        txtPromedio.setBounds(79, 40, 87, 23);
        getContentPane().add(txtPromedio);

        btnProcesar = new JButton("Procesar");
        btnProcesar.setBounds(305, 15, 101, 23);
        btnProcesar.addActionListener(this);
        getContentPane().add(btnProcesar);

        btnBorrar = new JButton("Borrar");
        btnBorrar.setBounds(305, 39, 101, 23);
        btnBorrar.addActionListener(this);
        getContentPane().add(btnBorrar);

        choCategoria = new JComboBox();
        choCategoria.setBounds(79, 15, 87, 23);
        choCategoria.addItem("A");
        choCategoria.addItem("B");
        choCategoria.addItem("C");
        choCategoria.addItem("D");
        getContentPane().add(choCategoria);

        lblCategoria = new JLabel("Categoría");
        lblCategoria.setBounds(15, 15, 61, 23);
        getContentPane().add(lblCategoria);

        lblPromedio = new JLabel("Promedio");
        lblPromedio.setBounds(15, 40, 56, 23);
        getContentPane().add(lblPromedio);

        txtS = new JTextArea();
        txtS.setFont(new Font("monospaced", 0, 12));

        scpScroll = new JScrollPane(txtS);
        scpScroll.setBounds(15, 75, 394, 75);
        getContentPane().add(scpScroll);
    }
    // -----
    // Procesa eventos de tipo ActionEvent
    public void actionPerformed( ActionEvent e ){
        if( e.getSource() == btnProcesar )
            procesar();
        if( e.getSource() == btnBorrar )
            borrar();
    }
}

```

```

// -----
// Método del botón Procesar
void procesar(){
    ingresarDatos();
    calcularPensionActual();
    calcularDescuento();
    calcularNuevaPension();
    mostrarResultados();
}
// -----
// Método del botón Borrar
void borrar(){
    txtS.setText("");
    txtPromedio.setText("");
    txtPromedio.requestFocus();
}
// -----
// Ingresa datos
void ingresarDatos(){
    categoria = choCategoria.getSelectedIndex();
    promedio = Double.parseDouble(txtPromedio.getText());
}
// -----
// Calcula la pensión actual
void calcularPensionActual(){
    if( categoria == 0 )
        actualpen = 550;
    else if( categoria == 1 )
        actualpen = 500;
    else if( categoria == 2 )
        actualpen = 460;
    else
        actualpen = 400;
}
// -----
// Calcula el descuento
void calcularDescuento(){
    if( promedio <= 13.99 )
        descuento = 0;
    else if( promedio <= 15.99 )
        descuento = 0.10*actualpen;
    else if( promedio <= 17.99 )
        descuento = 0.12*actualpen;
    else
        descuento = 0.15*actualpen;
}
// -----
// Calcula la nueva pensión
void calcularNuevaPension(){
    nuevapen = actualpen - descuento;
}
// -----
// Muestra resultados
void mostrarResultados(){
    txtS.setText("");
    imprimir("Pensión actual : " + actualpen);
    imprimir("Descuento      : " + descuento);
    imprimir("Nueva pensión  : " + nuevapen);
}
// -----
// Imprime una cadena de texto incluyendo un salto de línea al final
void imprimir(String cad){
    txtS.append(cad + "\n");
}
}

```

Problema 7.2

Una empresa evalúa a sus empleados bajo dos criterios: puntualidad y rendimiento. En cada caso el empleado recibe un puntaje que va de 1 a 10, de acuerdo a los siguientes criterios:

Puntaje por puntualidad:- está en función a los minutos de tardanza de acuerdo a la siguiente tabla:

Minutos de tardanza	Puntaje
0	10
1 a 2	8
3 a 5	6
6 a 9	4
Más de 9	0

Puntaje por rendimiento:- está en función a la cantidad de observaciones efectuadas al empleado por no cumplir sus obligaciones de acuerdo a la siguiente tabla:

Observaciones efectuadas	Puntaje
0	10
1	8
2	5
3	1
Más de 3	0

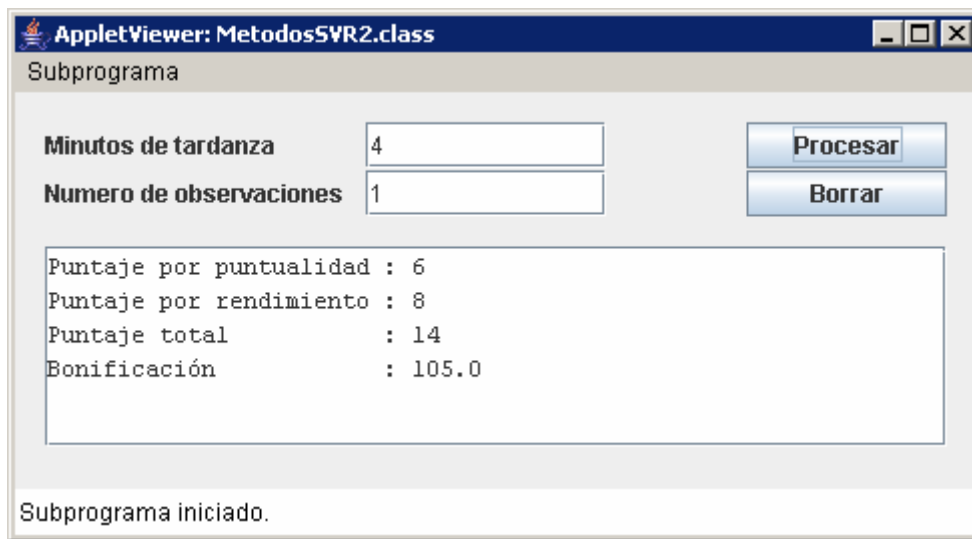
El puntaje total del empleado es la suma del puntaje por puntualidad más el puntaje por rendimiento. En base al puntaje total, el empleado recibe una bonificación anual de acuerdo a la siguiente tabla:

Puntaje total	Bonificación
Menos de 11	S/. 2.5 por punto
11 a 13	S/. 5.0 por punto
14 a 16	S/. 7.5 por punto
17 a 19	S/. 10.0 por punto
20	S/. 12.5 por punto

Diseñe un programa que determine el puntaje por puntualidad, el puntaje por rendimiento, el puntaje total y la bonificación que le corresponden a un empleado de la empresa.

Declare todas las variables como globales y use métodos tipo void.

Programa



```
import java.awt.event.*;
import java.awt.*;
import javax.swing.*;

public class MetodosSVR2 extends JApplet implements ActionListener{
    // Declaración de variables globales para la GUI
    JButton btnProcesar, btnBorrar;
    JTextArea txtS;
    JLabel lblMinutosTar, lblNumeroObs;
    JTextField txtMinutosTar, txtNumeroObs;
    JScrollPane scpScroll;

    // Declaración de variables globales para el algoritmo
    int minutosTar, numeroObs, puntajePun, puntajeRen, puntajeTot;
    double bonificacion;

    // -----
    // Crea la interfaz gráfica de usuario
    public void init() {
        getContentPane().setLayout(null);

        lblMinutosTar = new JLabel("Minutos de tardanza");
        lblMinutosTar.setBounds(15, 15, 160, 23);
        getContentPane().add(lblMinutosTar);

        lblNumeroObs = new JLabel("Numero de observaciones");
        lblNumeroObs.setBounds(15, 39, 160, 23);
        getContentPane().add(lblNumeroObs);

        txtMinutosTar = new JTextField();
        txtMinutosTar.setBounds(175, 15, 120, 23);
        getContentPane().add(txtMinutosTar);

        txtNumeroObs = new JTextField();
        txtNumeroObs.setBounds(175, 39, 120, 23);
        getContentPane().add(txtNumeroObs);

        btnProcesar = new JButton("Procesar");
        btnProcesar.setBounds(365, 15, 100, 23);
        btnProcesar.addActionListener(this);
        getContentPane().add(btnProcesar);

        btnBorrar = new JButton("Borrar");
        btnBorrar.setBounds(365, 39, 100, 23);
        btnBorrar.addActionListener(this);
    }
}
```

```

        getContentPane().add(btnBorrar);

        txtS = new JTextArea();
        txtS.setFont(new Font("monospaced", 0, 12));

        scpScroll = new JScrollPane(txtS);
        scpScroll.setBounds(15, 77, 450, 100);
        getContentPane().add(scpScroll);
    }
    // -----
    // Procesa eventos de tipo(ActionEvent)
    public void actionPerformed( ActionEvent e ){
        if(e.getSource() == btnProcesar)
            procesar();
        if(e.getSource() == btnBorrar)
            borrar();
    }
    // -----
    // Método del botón Procesar
    void procesar(){
        ingresarDatos();
        determinarPuntajePuntualidad();
        determinarPuntajeRendimiento();
        determinarPuntajeTotal();
        determinarBonificacion();
        mostrarResultados();
    }
    // -----
    // Método del botón Borrar
    void borrar(){
        txtMinutosTar.setText("");
        txtNumeroObs.setText("");
        txtS.setText("");
        txtMinutosTar.requestFocus();
    }
    // -----
    // Efectúa la entrada de datos
    void ingresarDatos(){
        minutosTar = Integer.parseInt(txtMinutosTar.getText());
        numeroObs = Integer.parseInt(txtNumeroObs.getText());
    }
    // -----
    // Determina el puntaje por puntualidad
    void determinarPuntajePuntualidad(){
        if(minutosTar == 0)
            puntajePun = 10;
        else if(minutosTar <= 2)
            puntajePun = 8;
        else if(minutosTar <= 5)
            puntajePun = 6;
        else if(minutosTar <= 9)
            puntajePun = 4;
        else
            puntajePun = 0;
    }
    // -----
    // Determina el puntaje por rendimiento
    void determinarPuntajeRendimiento(){
        if(numeroObs == 0)
            puntajeRen = 10;
        else if(numeroObs == 1)
            puntajeRen = 8;
        else if(numeroObs == 2)
            puntajeRen = 5;
        else if(numeroObs == 3)
            puntajeRen = 1;
        else
            puntajeRen = 0;
    }

```

```

    }
    // -----
    // Determina el puntaje total
    void determinarPuntajeTotal() {
        puntajeTot = puntajePun + puntajeRen;
    }
    // -----
    // Determina la bonificación
    void determinarBonificacion() {
        if(puntajeTot < 11)
            bonificacion = 2.5*puntajeTot;
        else if(puntajeTot <= 13)
            bonificacion = 5.0*puntajeTot;
        else if(puntajeTot <= 16)
            bonificacion = 7.5*puntajeTot;
        else if(puntajeTot <= 19)
            bonificacion = 10.0*puntajeTot;
        else
            bonificacion = 12.5*puntajeTot;
    }
    // -----
    // Muestra los resultados
    void mostrarResultados() {
        txtS.setText("");
        imprimir("Puntaje por puntualidad : " + puntajePun);
        imprimir("Puntaje por rendimiento : " + puntajeRen);
        imprimir("Puntaje total : " + puntajeTot);
        imprimir("Bonificación : " + bonificacion);
    }
    // -----
    // Imprime una cadena de texto incluyendo un salto de línea al final
    void imprimir(String cad) {
        txtS.append(cad + "\n");
    }
}

```

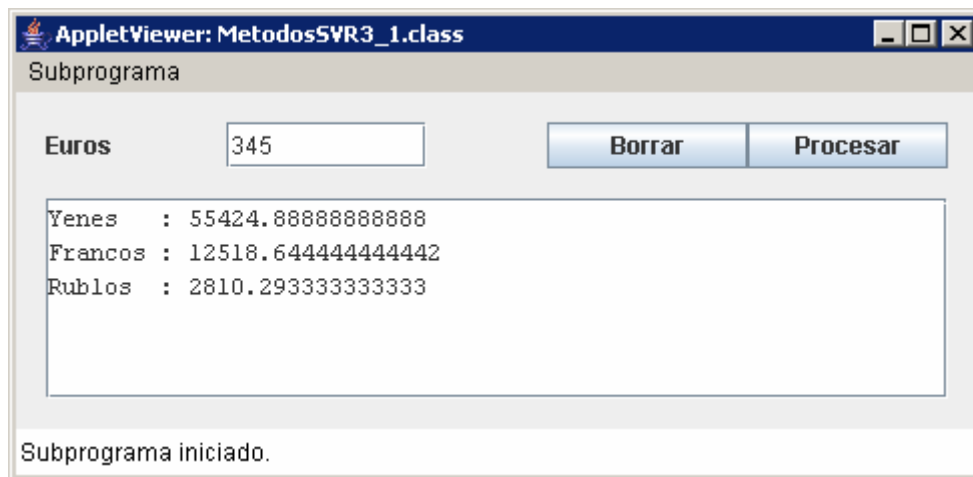
Problema 7.3

Diseñe un programa que permita convertir un monto de dinero dado en euros a sus equivalentes en yenes, francos y rublos. Considere que:

1 dólar = 0.6750 euros
 1 dólar = 108.44 yenes
 1 dólar = 24.493 francos
 1 dólar = 5.4984 rublos

Use métodos tipo void y considere todas las variables como globales.

Programa



```
import java.awt.event.*;
import java.awt.*;
import javax.swing.*;
```

```
public class MetodosSVR3_1 extends JApplet implements ActionListener{
    // Declaración de variables globales para la GUI
    JButton btnProcesar, btnBorrar;
    JTextArea txtS;
    JLabel lblEuros;
    JTextField txtEuros;
    JScrollPane scpScroll;

    // Declaración de variables globales para el algoritmo
    double meuros, myenes, mfrancos, mrublos;

    // -----
    // Crea la interfaz gráfica de usuario
    public void init() {
        getContentPane().setLayout(null);

        lblEuros = new JLabel("Euros");
        lblEuros.setBounds(15, 15, 90, 23);
        getContentPane().add(lblEuros);

        txtEuros = new JTextField();
        txtEuros.setBounds(105, 15, 100, 23);
        getContentPane().add(txtEuros);

        btnProcesar = new JButton("Procesar");
        btnProcesar.setBounds(365, 15, 100, 23);
        btnProcesar.addActionListener(this);
        getContentPane().add(btnProcesar);

        btnBorrar = new JButton("Borrar");
        btnBorrar.setBounds(265, 15, 100, 23);
        btnBorrar.addActionListener(this);
        getContentPane().add(btnBorrar);

        txtS = new JTextArea();
        txtS.setFont(new Font("monospaced", 0, 12));

        scpScroll = new JScrollPane(txtS);
        scpScroll.setBounds(15, 53, 450, 100);
        getContentPane().add(scpScroll);
    }
    // -----
    // Procesa eventos de tipo(ActionEvent)
    public void actionPerformed( ActionEvent e ){
        if(e.getSource() == btnProcesar)
            procesar();
    }
}
```



```

        if(e.getSource() == btnBorrar)
            borrar();
    }
    // -----
    // Método del botón Procesar
    void procesar(){
        ingresarDatos();
        convertirAYenes();
        convertirAFrancos();
        convertirARublos();
        mostrarResultados();
    }
    // -----
    // Método del botón Borrar
    void borrar(){
        txtEuros.setText("");
        txtS.setText("");
        txtEuros.requestFocus();
    }
    // -----
    // Ingresa datos
    void ingresarDatos(){
        meuros = Double.parseDouble(txtEuros.getText());
    }
    // -----
    // Convierte a yenes
    void convertirAYenes(){
        myenes = (meuros/0.6750)*108.44;
    }
    // -----
    // Convierte a francos
    void convertirAFrancos(){
        mfrancos = (meuros/0.6750)*24.493;
    }
    // -----
    // Convierte a rublos
    void convertirARublos(){
        mrublos = (meuros/0.6750)*5.4984;
    }
    // -----
    // Muestra resultados
    void mostrarResultados(){
        txtS.setText("");
        imprimir("Yenes      : " + myenes);
        imprimir("Francos    : " + mfrancos);
        imprimir("Rublos     : " + mrublos);
    }
    // -----
    // Imprime una cadena de texto incluyendo un salto de línea al final
    void imprimir(String cad){
        txtS.append(cad + "\n");
    }
}

```

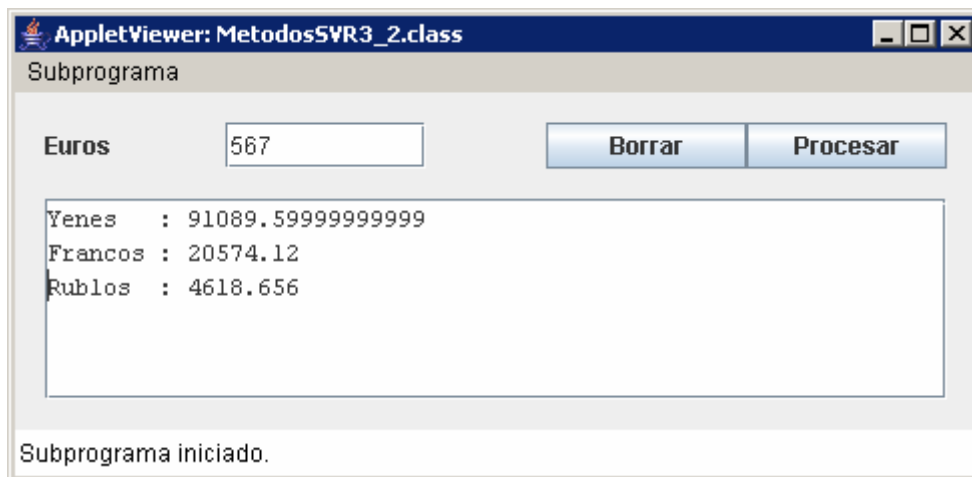
Problema 7.4

Diseñe un programa que permita convertir un monto de dinero dado en euros a sus equivalentes en yenes, francos y rublos. Considere que:

1 dólar = 0.6750 euros
 1 dólar = 108.44 yenes
 1 dólar = 24.493 francos
 1 dólar = 5.4984 rublos

Use métodos tipo void y considere todas las variables como locales.

Programa



```
import java.awt.event.*;
import java.awt.*;
import javax.swing.*;
```

```
public class MetodosSVR3_2 extends JApplet implements ActionListener{
    // Declaración de variables globales
    JButton btnProcesar, btnBorrar;
    JTextArea txtS;
    JLabel lblEuros;
    JTextField txtEuros;
    JScrollPane scpScroll;

    // -----
    // Crea la interfaz gráfica de usuario
    public void init() {
        getContentPane().setLayout(null);

        lblEuros = new JLabel("Euros");
        lblEuros.setBounds(15, 15, 90, 23);
        getContentPane().add(lblEuros);

        txtEuros = new JTextField();
        txtEuros.setBounds(105, 15, 100, 23);
        getContentPane().add(txtEuros);

        btnProcesar = new JButton("Procesar");
        btnProcesar.setBounds(365, 15, 100, 23);
        btnProcesar.addActionListener(this);
        getContentPane().add(btnProcesar);

        btnBorrar = new JButton("Borrar");
        btnBorrar.setBounds(265, 15, 100, 23);
        btnBorrar.addActionListener(this);
        getContentPane().add(btnBorrar);

        txtS = new JTextArea();
        txtS.setFont(new Font("monospaced", 0, 12));

        scpScroll = new JScrollPane(txtS);
        scpScroll.setBounds(15, 53, 450, 100);
        getContentPane().add(scpScroll);
    }
    // -----
    // Procesa eventos de tipo(ActionEvent)
    public void actionPerformed( ActionEvent e ){
        if(e.getSource() == btnProcesar)
            procesar();
        if(e.getSource() == btnBorrar)
```

```

        borrar();
    }
    // -----
    // Método del botón Procesar
    void procesar(){
        // Declaración de variables locales
        double meuros;

        // Entrada de datos
        meuros = Double.parseDouble(txtEuros.getText());

        // Invocación de métodos
        txtS.setText("");
        convertirAYenes(meuros);
        convertirAFrancos(meuros);
        convertirARublos(meuros);
    }
    // -----
    // Convierte a yenes
    void convertirAYenes(double xeuros){
        double myenes;
        myenes = (xeuros/0.6750)*108.44;
        imprimir("Yenes    : " + myenes + "\n");
    }
    // -----
    // Convierte a francos
    void convertirAFrancos(double xeuros){
        double mfrancos;
        mfrancos = (xeuros/0.6750)*24.493;
        imprimir("Francos  : " + mfrancos + "\n");
    }
    // -----
    // Convierte a rublos
    void convertirARublos(double xeuros){
        double mrublos;
        mrublos = (xeuros/0.6750)*5.4984;
        imprimir("Rublos   : " + mrublos + "\n");
    }
    // -----
    // Método del botón Borrar
    void borrar(){
        txtEuros.setText("");
        txtS.setText("");
        txtEuros.requestFocus();
    }
    // -----
    // Imprime una cadena de texto incluyendo un salto de línea al final
    void imprimir(String cad){
        txtS.append(cad + "\n");
    }
}

```

Problema 7.5

Diseñe un programa que determine el área lateral, el área total y el área de la base de un cilindro. Considere que:

$$areabase = \pi r^2$$

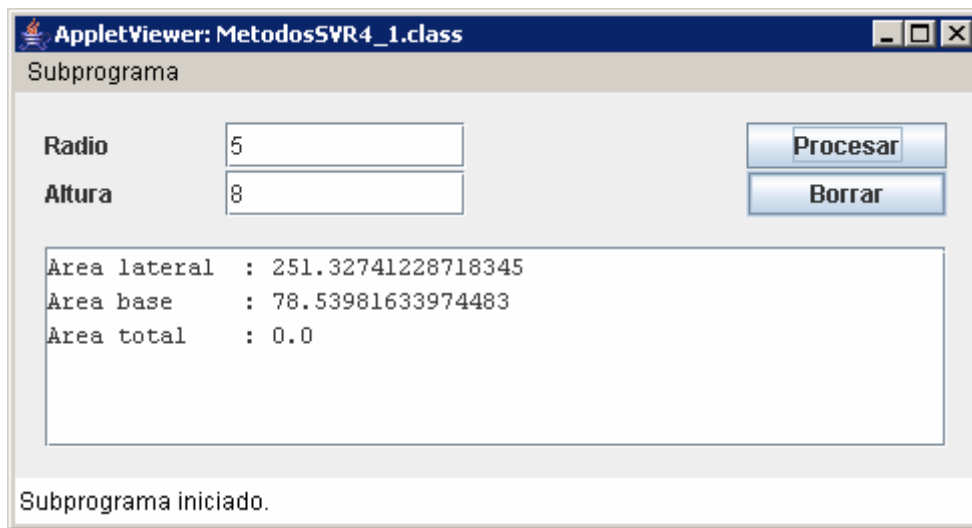
$$arealateral = 2\pi r h$$

$$areatotal = 2\pi r(r + h)$$

Donde r es el radio y h la altura.

Declare todas las variables como globales y use métodos tipo void.

Programa



```
import java.awt.event.*;
import java.awt.*;
import javax.swing.*;
```

```
public class MetodosSVR4_1 extends JApplet implements ActionListener{
    // Declaración de variables globales para la GUI
    JButton btnProcesar, btnBorrar;
    JTextArea txtS;
    JLabel lblRadio, lblAltura;
    JTextField txtRadio, txtAltura;
    JScrollPane scpScroll;

    // Declaración de variables globales para el algoritmo
    double radio, altura, arealat, areabas, areatot;

    // -----
    // Crea la interfaz gráfica de usuario
    public void init() {
        getContentPane().setLayout(null);

        lblRadio = new JLabel("Radio");
        lblRadio.setBounds(15, 15, 90, 23);
        getContentPane().add(lblRadio);

        lblAltura = new JLabel("Altura");
        lblAltura.setBounds(15, 39, 90, 23);
        getContentPane().add(lblAltura);

        txtRadio = new JTextField();
        txtRadio.setBounds(105, 15, 120, 23);
        getContentPane().add(txtRadio);

        txtAltura = new JTextField();
        txtAltura.setBounds(105, 39, 120, 23);
        getContentPane().add(txtAltura);

        btnProcesar = new JButton("Procesar");
        btnProcesar.setBounds(365, 15, 100, 23);
        btnProcesar.addActionListener(this);
        getContentPane().add(btnProcesar);

        btnBorrar = new JButton("Borrar");
        btnBorrar.setBounds(365, 39, 100, 23);
        btnBorrar.addActionListener(this);
        getContentPane().add(btnBorrar);
    }
}
```

```

        txtS = new JTextArea();
        txtS.setFont(new Font("monospaced", 0, 12));

        scpScroll = new JScrollPane(txtS);
        scpScroll.setBounds(15, 77, 450, 100);
        getContentPane().add(scpScroll);
    }
    // -----
    // Procesa eventos de tipo ActionEvent
    public void actionPerformed( ActionEvent e ){
        if(e.getSource() == btnProcesar)
            procesar();
        if(e.getSource() == btnBorrar)
            borrar();
    }
    // -----
    // Método para el botón Procesar
    void procesar(){
        ingresarDatos();
        calcularAreaBase();
        calcularAreaLateral();
        mostrarResultados();
    }
    // -----
    // Método para el botón Borrar
    void borrar(){
        txtRadio.setText("");
        txtAltura.setText("");
        txtS.setText("");
        txtRadio.requestFocus();
    }
    // -----
    // Ingresar datos
    void ingresarDatos(){
        radio = Double.parseDouble(txtRadio.getText());
        altura = Double.parseDouble(txtAltura.getText());
    }
    // -----
    // Calcula el área de la base
    void calcularAreaBase(){
        areabas = Math.PI*radio*radio;
    }
    // -----
    // Calcula el área lateral
    void calcularAreaLateral(){
        arealat = 2*Math.PI*radio*altura;
    }
    // -----
    // Calcula el área total
    void calcularAreaTotal(){
        areatot = 2*Math.PI*radio*(radio+ altura);
    }
    // -----
    // Muestra resultados
    void mostrarResultados(){
        txtS.setText("");
        imprimir("Area lateral   : " + arealat);
        imprimir("Area base      : " + areabas);
        imprimir("Area total    : " + areatot);
    }
    // -----
    // Imprime una línea de texto con un salto de línea al final
    void imprimir( String cad ) {
        txtS.append(cad + "\n");
    }
}

```

Problema 7.6

Diseñe un programa que determine el área lateral, el área total y el área de la base de un cilindro. Considere que:

$$areabase = \pi r^2$$

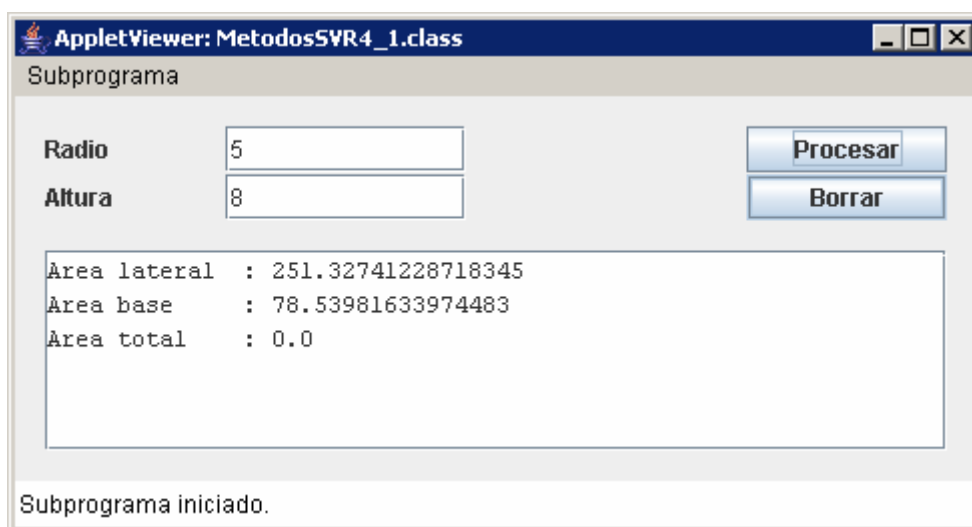
$$arealateral = 2\pi r h$$

$$areatotal = 2\pi r(r + h)$$

Donde r es el radio y h la altura.

Declare todas las variables como locales y use métodos tipo void.

Programa



```
import java.awt.event.*;
import java.awt.*;
import javax.swing.*;

public class MetodosSVR4_2 extends JApplet implements ActionListener{
    // Declaración de variables globales para la GUI
    JButton btnProcesar, btnBorrar;
    JTextArea txtS;
    JLabel lblRadio, lblAltura;
    JTextField txtRadio, txtAltura;
    JScrollPane scpScroll;

    // -----
    // Crea la interfaz gráfica de usuario
    public void init() {
        getContentPane().setLayout(null);

        lblRadio = new JLabel("Radio");
        lblRadio.setBounds(15, 15, 90, 23);
        getContentPane().add(lblRadio);

        lblAltura = new JLabel("Altura");
        lblAltura.setBounds(15, 39, 90, 23);
        getContentPane().add(lblAltura);

        txtRadio = new JTextField();
        txtRadio.setBounds(105, 15, 120, 23);
```

```

        getContentPane().add(txtRadio);

        txtAltura = new JTextField();
        txtAltura.setBounds(105, 39, 120, 23);
        getContentPane().add(txtAltura);

        btnProcesar = new JButton("Procesar");
        btnProcesar.setBounds(365, 15, 100, 23);
        btnProcesar.addActionListener(this);
        getContentPane().add(btnProcesar);

        btnBorrar = new JButton("Borrar");
        btnBorrar.setBounds(365, 39, 100, 23);
        btnBorrar.addActionListener(this);
        getContentPane().add(btnBorrar);

        txtS = new JTextArea();
        txtS.setFont(new Font("monospaced", 0, 12));

        scpScroll = new JScrollPane(txtS);
        scpScroll.setBounds(15, 77, 450, 100);
        getContentPane().add(scpScroll);
    }
    // -----
    // Procesa eventos de tipo ActionEvent
    public void actionPerformed( ActionEvent e ){
        if(e.getSource() == btnProcesar)
            procesar();
        if(e.getSource() == btnBorrar)
            borrar();
    }
    // -----
    // Método para el botón Procesar
    void procesar(){

        // Declaración de variables locales
        double radio, altura;

        // Entrada de datos
        radio = Double.parseDouble(txtRadio.getText());
        altura = Double.parseDouble(txtAltura.getText());

        // Invocación de métodos
        txtS.setText("");
        calcularAreaBase(radio);
        calcularAreaLateral(radio, altura);
        calcularAreaTotal(radio, altura);
    }
    // -----
    // Método para el botón Borrar
    void borrar(){
        txtRadio.setText("");
        txtAltura.setText("");
        txtS.setText("");
        txtRadio.requestFocus();
    }
    // -----
    // Calcula el área de la base
    void calcularAreaBase(double radio){
        double areabas;
        areabas = Math.PI*radio*radio;
        imprimir("Area base      : " + areabas);
    }
    // -----
    // Calcula el área lateral
    void calcularAreaLateral(double radio, double altura){
        double arealat;
        arealat = 2*Math.PI*radio*altura;
    }

```

```

        imprimir("Area lateral  : " + arealat);
    }
    // -----
    // Calcula el área total
    void calcularAreaTotal(double radio, double altura){
        double areatot;
        areatot = 2*Math.PI*radio*(radio+ altura);
        imprimir("Area total    : " + areatot);
    }
    // -----
    // Imprime una línea de texto con un salto de línea al final
    void imprimir( String cad ) {
        txtS.append(cad + "\n");
    }
}

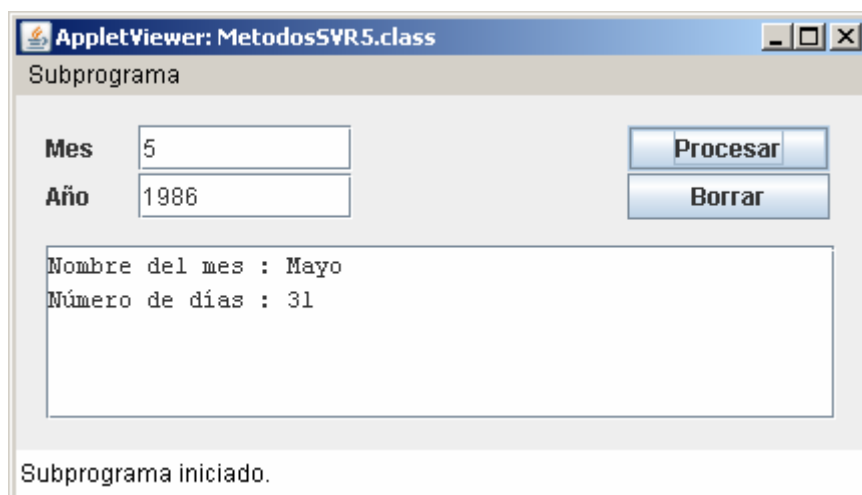
```

Problema 7.7

Diseñe un programa que determine la cantidad de días y el nombre de un mes conociendo los valores numéricos del mes y del año.

Declare todas las variables como locales y use métodos tipo void.

Programa



```

import java.awt.event.*;
import java.awt.*;
import javax.swing.*;

public class MetodosSVR5 extends JApplet implements ActionListener{
    // Declaración de variables globales para la GUI
    JButton btnProcesar, btnBorrar;
    JTextArea txtS;
    JLabel lblAño, lblMes;
    JTextField txtAño, txtMes;
    JScrollPane scpScroll;

    // -----
    // Crea la interfaz gráfica de usuario GUI
    public void init() {
        getContentPane().setLayout(null);

        txtMes = new JTextField();
        txtMes.setBounds(61, 15, 107, 23);
        getContentPane().add(txtMes);

        txtAño = new JTextField();

```



```

txtAño.setBounds(61, 39, 107, 23);
getContentPane().add(txtAño);

btnProcesar = new JButton("Procesar");
btnProcesar.setBounds(305, 15, 101, 23);
btnProcesar.addActionListener(this);
getContentPane().add(btnProcesar);

btnBorrar = new JButton("Borrar");
btnBorrar.setBounds(305, 39, 101, 23);
btnBorrar.addActionListener(this);
getContentPane().add(btnBorrar);

lblAño = new JLabel("Año");
lblAño.setBounds(15, 39, 42, 23);
getContentPane().add(lblAño);

lblMes = new JLabel("Mes");
lblMes.setBounds(15, 15, 42, 23);
getContentPane().add(lblMes);

txtS = new JTextArea();
txtS.setFont(new Font("monospaced", 0, 12));

scpScroll = new JScrollPane(txtS);
scpScroll.setBounds(15, 75, 394, 87);
getContentPane().add(scpScroll);
}
// -----
// Procesa eventos de tipo ActionEvent
public void actionPerformed( ActionEvent e ){
    if(e.getSource() == btnProcesar)
        procesar();
    if(e.getSource() == btnBorrar)
        borrar();
}
// -----
// Método del botón Procesar
void procesar(){
    // Declaración de variables locales
    int mes, año;

    // Entrada de datos
    mes = Integer.parseInt(txtMes.getText());
    año = Integer.parseInt(txtAño.getText());

    // Invocación de métodos
    txtS.setText("");
    determinaNombreMes(mes);
    determinaNumeroDiasMes(mes, año);
}
// -----
// Método del botón Borrar
void borrar(){
    txtS.setText("");
    txtMes.setText("");
    txtAño.setText("");
    txtMes.requestFocus();
}
// -----
// Determina el nombre del mes
void determinaNombreMes(int mes){
    // Declaración de variables locales
    String nombre;

    // Determina el nombre del mes
    switch(mes){
        case 1 :

```

```

        nombre = "Enero";
        break;
    case 2 :
        nombre = "Febrero";
        break;
    case 3 :
        nombre = "Marzo";
        break;
    case 4 :
        nombre = "Abril";
        break;
    case 5 :
        nombre = "Mayo";
        break;
    case 6 :
        nombre = "Junio";
        break;
    case 7 :
        nombre = "Julio";
        break;
    case 8 :
        nombre = "Agosto";
        break;
    case 9 :
        nombre = "Setiembre";
        break;
    case 10:
        nombre = "Octubre";
        break;
    case 11:
        nombre = "Noviembre";
        break;
    default:
        nombre = "Diciembre";
}

// Muestra el nombre del mes
imprimir("Nombre del mes : " + nombre);
}
// -----
// Determina el número de días del mes
void determinaNumeroDiasMes(int mes, int año){
    // Declaración de variables locales
    int dias;

    // Determina el número de días del mes
    switch(mes){
        case 1:
        case 3:
        case 5:
        case 7:
        case 8:
        case 10:
        case 12:
            dias = 31;
            break;
        case 4:
        case 6:
        case 9:
        case 11:
            dias = 30;
            break;
        default:
            if((año%4 == 0) && ((año%100 != 0) || (año%400 == 0)))
                dias = 29;
            else
                dias = 28;
    }
}

```

```

        // Muestra el número de días del mes
        imprimir("Número de días : " + dias);
    }
    // -----
    // Imprime una cadena de texto incluyendo un salto de línea al final
    void imprimir(String cad){
        txtS.append(cad + "\n");
    }
}

```

Problemas propuestos

Actividad

1. Una tienda vende tres tipos de productos a los precios unitarios dados en la siguiente tabla:

Producto	Precio
P1	S/. 15.0
P2	S/. 17.5
P3	S/. 20.0

Como oferta la tienda ofrece un regalo de acuerdo a la siguiente tabla:

Unidades adquiridas	Regalo
1 a 25	un lapicero
26 a 50	un cuaderno
Más de 50	una agenda

Diseñe un programa que determine el importe a pagar y el regalo para un cliente de la tienda.

Solución 1:- Declare todas las variables como globales y use métodos tipo void.

Solución 2:- Declare todas las variables como locales y use métodos tipo void.

2. Determine que se imprime al ejecutar el método **procesar** correspondiente a un botón de un programa:

```

void procesar(){
    int a = 5, b = 10, c = 5;
    metodo1(a, b, c+1);
    metodo2(c);
}

void metodo1(int p, int q, int r){
    int s;
    s = p+q+r;
    txtS.append(s + "\n");
}

```

```

void metodo2(int c){
    txtS.append(c + "\n");
    metodo3(c+2);
}

void metodo3(int x){
    txtS.append(x + "\n");
}

```

Autoevaluación

1. Diseñe un programa que lea un número entero y determine:

- El valor absoluto del número.
- El signo del número entre: positivo, negativo o neutro.

Declare todas las variables como locales y use métodos tipo void diferentes para cada uno de los cálculos.

2. Una empresa calcula el sueldo bruto de sus trabajadores multiplicando las horas trabajadas por una tarifa horaria que depende de la categoría del trabajador de acuerdo a la siguiente tabla:

Categoría	Tarifa
A	S/. 21.0
B	S/. 19.5
C	S/. 17.0
D	S/. 15.5

Por ley, todo trabajador se somete a un porcentaje de descuento del sueldo bruto: 20% si el sueldo bruto es mayor que S/. 2500 y 15% en caso contrario.

Diseñe un programa que determine el sueldo bruto, el descuento y el sueldo neto que le corresponden a un trabajador de la empresa.

Declare todas las variables como globales y use métodos tipo void diferentes para cada uno de los cálculos.

3. Determine que se imprime al ejecutar el método **procesar** correspondiente a un botón de un programa:

```

void procesar(){
    metodo1(20, 15);
    metodo4(10);
}

void metodo1(int p, int q){
    metodo2(2*p);
    metodo3(2*q);
}

void metodo2(int n){
    txtS.append(n + "\n");
}

void metodo3(int m){

```

```
        txtS.append(m + "\n");
    }

    void metodo4(int r){
        int aux = r*r*r;
        txtS.append(aux + "\n");
    }
```

4. Determine que se imprime al ejecutar el método **procesar** correspondiente a un botón de un programa:

```
void procesar(){
    int a = 135, b = 267;
    metodo1(a, b);
    metodo2(b);
}

void metodo1(int n, int m){
    int r;
    r = n%10 + m%10;
    txtS.append(r + "\n");
}

void metodo2(int c){
    txtS.append(c + "\n");
}
```

Para recordar

- Una variable local sólo puede ser utilizada dentro del método donde fue declarada.
- Una variable global puede ser utilizada en todo el programa.
- Un método tipo void es un método que puede recibir datos de entrada a través de variables locales al método conocidas como parámetros.

**UNIDAD DE
APRENDIZAJE****4****SEMANA****11**

MÉTODOS CON VALOR DE RETORNO

LOGRO DE LA UNIDAD DE APRENDIZAJE

Al finalizar la unidad, los alumnos, utilizando descomposición en módulos mediante métodos tipo void y con valor de retorno y, los demás conceptos aprendidos en el curso, diseñarán programas en Java que resuelvan problemas planteados.

CONTENIDO

1. Métodos con valor de retorno

ACTIVIDADES

Los alumnos desarrollan programas usando métodos con valor de retorno y tipo void.

1. METODOS CON VALOR DE RETORNO

Un método con valor de retorno es un módulo de programa que puede recibir datos de entrada a través de variables locales denominadas parámetros y que retorna un resultado al punto donde es invocado. Este tipo de método se utiliza para efectuar cualquier tipo de proceso que produzca un resultado

Estos métodos pueden dividirse a su vez en dos tipos:

- Métodos con valor de retorno sin parámetros.
- Métodos con valor de retorno con parámetros.

1.1 Métodos con valor de retorno, sin parámetros

Este tipo de métodos no reciben datos de entrada; pero, retornan un resultado al punto donde son invocados.



Figura 8.1 Método con valor de retorno sin parámetros

Definición

Este tipo de método se define de la siguiente manera:

```
tipo nombre() {
    Declaración de variables locales
    Cuerpo del método
    return valor;
}
```

Donde:

nombre : Es el nombre del método
valor : Es el valor a ser retornado por el método
tipo : Es el tipo del valor de retorno.

Llamada

Este tipo de método se invoca de la siguiente manera::

```
variable = nombre();
```

Donde:

variable : Es la variable que recibe el valor retornado por el método.
nombre : Es el nombre del método invocado.

1.2 Métodos con valor de retorno con parámetros

Estos métodos reciben datos de entrada a través de parámetros y retornan un resultado al punto de su invocación.

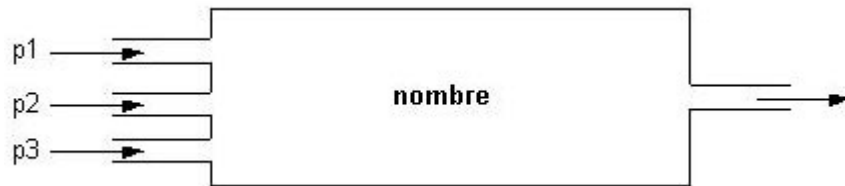


Figura 8.2 Método con valor de retorno y con parámetros

Definición

Este tipo de método se define de la siguiente manera:

```
tipo nombre( tipo1 p1, tipo2 p2, tipo3 p3, . . . ){
    Declaración de variables locales
    Cuerpo del método
    return valor;
}
```

Donde:

nombre : Es el nombre del método
tipo : Es el tipo del valor de retorno.
p1, p2, p3, ... : Son los nombres de los parámetros
tipo1, tipo2, tipo3, ... : Son los tipos de datos de los parámetros
valor : Es el valor de retorno

Llamada

Este tipo de método se invoca de la siguiente manera:

```
variable = nombre(v1, v2, v3, ...);
```

Donde:

nombre : Es el nombre del método invocado
variable : Es la variable que recibe el valor de retorno
v1, v2, v3, ... : Son los valores dados a los parámetros

Problema 8.1

En una universidad, los alumnos están categorizados en cuatro categorías. A cada categoría le corresponde una pensión mensual distinta dada en la siguiente tabla:

Categoría	Pensión
A	S/. 550
B	S/. 500
C	S/. 460
D	S/. 400

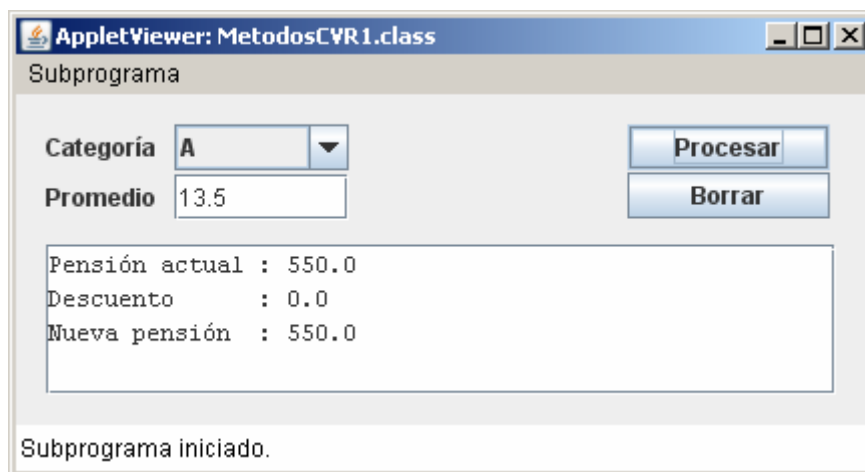
Semestralmente, la universidad efectúa rebajas en las pensiones de sus estudiantes a partir del segundo ciclo en base al promedio ponderado del ciclo anterior en porcentajes dados en la tabla siguiente:

Promedio	Descuento
0 a 13.99	No hay descuento
14.00 a 15.99	10 %
16.00 a 17.99	12 %
18.00 a 20.00	15 %

Diseñe un programa que determine cuánto de rebaja recibirá un estudiante sobre su pensión actual y a cuánto asciende su nueva pensión.

Declare todas las variables como locales y use métodos con valor de retorno para cada una de las entradas y para cada uno de los cálculos.

Programa 1



```
import java.awt.event.*;
import java.awt.*;
import javax.swing.*;
```

```
public class MetodosCVR1 extends JApplet implements ActionListener{
    // Declaración de variables globales
    JButton btnProcesar, btnBorrar;
    JTextArea txtS;
    JComboBox choCategoria;
    JLabel lblCategoria, lblPromedio;
    JTextField txtPromedio;
    JScrollPane scpScroll;

    // -----
    // Crea la interfaz gráfica de usuario GUI
    public void init() {
        getContentPane().setLayout(null);
        setBackground(new Color(214,211,206));

        txtPromedio = new JTextField();
        txtPromedio.setBounds(79, 40, 87, 23);
        getContentPane().add(txtPromedio);

        btnProcesar = new JButton("Procesar");
        btnProcesar.setBounds(305, 15, 101, 23);
```

```

        btnProcesar.addActionListener(this);
        getContentPane().add(btnProcesar);

        btnBorrar = new JButton("Borrar");
        btnBorrar.setBounds(305, 39, 101, 23);
        btnBorrar.addActionListener(this);
        getContentPane().add(btnBorrar);

        choCategoria = new JComboBox();
        choCategoria.setBounds(79, 15, 87, 23);
        choCategoria.addItem("A");
        choCategoria.addItem("B");
        choCategoria.addItem("C");
        choCategoria.addItem("D");
        getContentPane().add(choCategoria);

        lblCategoria = new JLabel("Categoría");
        lblCategoria.setBounds(15, 15, 61, 23);
        getContentPane().add(lblCategoria);

        lblPromedio = new JLabel("Promedio");
        lblPromedio.setBounds(15, 40, 56, 23);
        getContentPane().add(lblPromedio);

        txtS = new JTextArea();
        txtS.setFont(new Font("monospaced", 0, 12));

        scpScroll = new JScrollPane(txtS);
        scpScroll.setBounds(15, 75, 394, 75);
        getContentPane().add(scpScroll);
    }
    // -----
    // Procesa eventos de tipo ActionEvent
    public void actionPerformed( ActionEvent e ){
        if( e.getSource() == btnProcesar )
            procesar();
        if( e.getSource() == btnBorrar )
            borrar();
    }
    // -----
    // Método del botón Procesar
    void procesar(){
        // Declaración de variables locales
        int categoria;
        double actualpen, nuevapen, descuento, promedio;

        // Entrada de datos
        categoria = getCategoria();
        promedio = getPromedio();

        // Proceso de cálculo
        actualpen = calcularPensionActual(categoria);
        descuento = calcularDescuento(promedio, actualpen);
        nuevapen = calcularNuevaPension(actualpen, descuento);

        // Salida de resultados
        txtS.setText("");
        imprimir("Pensión actual : " + actualpen);
        imprimir("Descuento      : " + descuento);
        imprimir("Nueva pensión  : " + nuevapen);
    }
    // -----
    // Método del botón Borrar
    void borrar(){
        txtS.setText("");
        txtPromedio.setText("");
        txtPromedio.requestFocus();
    }
}

```

```

// -----
// Lee y retorna la categoría
int getCategory(){
    int categoria;
    categoria = choCategoria.getSelectedIndex();
    return categoria;
}
// -----
// Lee y retorna el promedio
double getPromedio(){
    double promedio;
    promedio = Double.parseDouble(txtPromedio.getText());
    return promedio;
}
// -----
// Calcula y retorna la pensión actual
double calcularPensionActual(int categoria){
    double actualpen;

    switch(categoria){
        case 0:
            actualpen = 550;
            break;
        case 1:
            actualpen = 500;
            break;
        case 2:
            actualpen = 460;
            break;
        default:
            actualpen = 400;
    }

    return actualpen;
}
// -----
// Cálcula el descuento
double calcularDescuento(double promedio, double actualpen){
    double descuento;

    if( promedio <= 13.99 )
        descuento = 0;
    else if( promedio <= 15.99 )
        descuento = 0.10*actualpen;
    else if( promedio <= 17.99 )
        descuento = 0.12*actualpen;
    else
        descuento = 0.15*actualpen;

    return descuento;
}
// -----
// Calcula la nueva pensión
double calcularNuevaPension(double actualpen, double descuento){
    double nuevapen;
    nuevapen = actualpen - descuento;
    return nuevapen;
}
// -----
// Imprime una cadena de texto incluyendo un salto de línea al final
void imprimir(String cad){
    txtS.append(cad + "\n");
}
}

```

Programa 2

```

import java.awt.event.*;
import java.awt.*;
import javax.swing.*;

public class MetodosCVR1 extends JApplet implements ActionListener{
    // Declaración de variables globales
    JButton btnProcesar, btnBorrar;
    JTextArea txtS;
    JComboBox choCategoria;
    JLabel lblCategoria, lblPromedio;
    JTextField txtPromedio;
    JScrollPane scpScroll;

    // -----
    // Crea la interfaz gráfica de usuario GUI
    public void init() {
        getContentPane().setLayout(null);
        setBackground(new Color(214,211,206));

        txtPromedio = new JTextField();
        txtPromedio.setBounds(79, 40, 87, 23);
        getContentPane().add(txtPromedio);

        btnProcesar = new JButton("Procesar");
        btnProcesar.setBounds(305, 15, 101, 23);
        btnProcesar.addActionListener(this);
        getContentPane().add(btnProcesar);

        btnBorrar = new JButton("Borrar");
        btnBorrar.setBounds(305, 39, 101, 23);
        btnBorrar.addActionListener(this);
        getContentPane().add(btnBorrar);

        choCategoria = new JComboBox();
        choCategoria.setBounds(79, 15, 87, 23);
        choCategoria.addItem("A");
        choCategoria.addItem("B");
        choCategoria.addItem("C");
        choCategoria.addItem("D");
        getContentPane().add(choCategoria);

        lblCategoria = new JLabel("Categoría");
        lblCategoria.setBounds(15, 15, 61, 23);
        getContentPane().add(lblCategoria);

        lblPromedio = new JLabel("Promedio");
        lblPromedio.setBounds(15, 40, 56, 23);
        getContentPane().add(lblPromedio);

        txtS = new JTextArea();
        txtS.setFont(new Font("monospaced", 0, 12));

        scpScroll = new JScrollPane(txtS);
        scpScroll.setBounds(15, 75, 394, 75);
        getContentPane().add(scpScroll);
    }
    // -----
    // Procesa eventos de tipo(ActionEvent)
    public void actionPerformed( ActionEvent e ){
        if( e.getSource() == btnProcesar )
            procesar();
        if( e.getSource() == btnBorrar )
            borrar();
    }
    // -----
    // Método del botón Procesar
    void procesar(){
        // Declaración de variables locales

```

```

    int categoria;
    double actualpen, nuevapen, descuento, promedio;

    // Entrada de datos
    categoria = getCategoria();
    promedio = getPromedio();

    // Proceso de cálculo
    actualpen = calcularPensionActual(categoria);
    descuento = calcularDescuento(promedio, actualpen);
    nuevapen = calcularNuevaPension(actualpen, descuento);

    // Salida de resultados
    txtS.setText("");
    imprimir("Pensión actual : " + actualpen);
    imprimir("Descuento      : " + descuento);
    imprimir("Nueva pensión  : " + nuevapen);
}
// -----
// Método del botón Borrar
void borrar(){
    txtS.setText("");
    txtPromedio.setText("");
    txtPromedio.requestFocus();
}
// -----
// Lee y retorna la categoría
int getCategoria(){
    return choCategoria.getSelectedIndex();
}
// -----
// Lee y retorna el promedio
double getPromedio(){
    return Double.parseDouble(txtPromedio.getText());
}
// -----
// Calcula y retorna la pensión actual
double calcularPensionActual(int categoria){
    switch(categoria){
        case 0 : return 550;
        case 1 : return 500;
        case 2 : return 460;
        default: return 400;
    }
}
// -----
// Calcula el descuento
double calcularDescuento(double promedio, double actualpen){
    if(promedio <= 13.99)
        return 0;
    else if(promedio <= 15.99)
        return 0.10*actualpen;
    else if(promedio <= 17.99)
        return 0.12*actualpen;
    else
        return 0.15*actualpen;
}
// -----
// Calcula la nueva pensión
double calcularNuevaPension(double actualpen, double descuento){
    return actualpen - descuento;
}
// -----
// Imprime una cadena de texto incluyendo un salto de línea al final
void imprimir(String cad){
    txtS.append(cad + "\n");
}
}

```

Problema 8.2

Una empresa evalúa a sus empleados bajo dos criterios: puntualidad y rendimiento. En cada caso el empleado recibe un puntaje que va de 1 a 10, de acuerdo a los siguientes criterios:

Puntaje por puntualidad:- está en función a los minutos de tardanza de acuerdo a la siguiente tabla:

Minutos de tardanza	Puntaje
0	10
1 a 2	8
3 a 5	6
6 a 9	4
Más de 9	0

Puntaje por rendimiento:- está en función a la cantidad de observaciones efectuadas al empleado por no cumplir sus obligaciones de acuerdo a la siguiente tabla:

Observaciones efectuadas	Puntaje
0	10
1	8
2	5
3	1
Más de 3	0

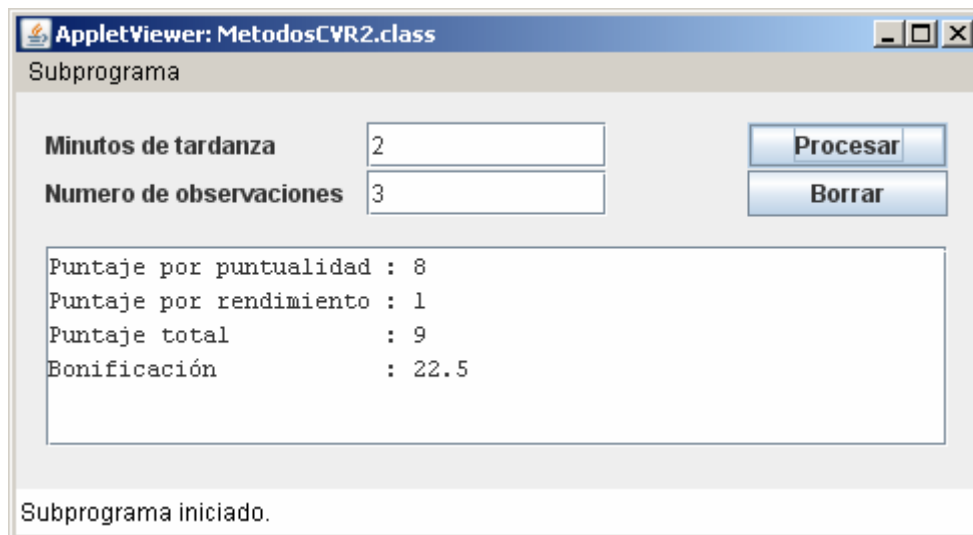
El puntaje total del empleado es la suma del puntaje por puntualidad más el puntaje por rendimiento. En base al puntaje total, el empleado recibe una bonificación anual de acuerdo a la siguiente tabla:

Puntaje total	Bonificación
Menos de 11	S/. 2.5 por punto
11 a 13	S/. 5.0 por punto
14 a 16	S/. 7.5 por punto
17 a 19	S/. 10.0 por punto
20	S/. 12.5 por punto

Diseñe un programa que determine el puntaje por puntualidad, el puntaje por rendimiento, el puntaje total y la bonificación que le corresponden a un empleado de la empresa.

Declare todas las variables como locales y use métodos con valor de retorno para cada una de las entradas y para cada uno de los cálculos.

Programa



```
import java.awt.event.*;
import java.awt.*;
import javax.swing.*;
```

```
public class MetodosCVR2 extends JApplet implements ActionListener{
    // Declaración de variables globales
    JButton btnProcesar, btnBorrar;
    JTextArea txtS;
    JLabel lblMinutosTar, lblNumeroObs;
    JTextField txtMinutosTar, txtNumeroObs;
    JScrollPane scpScroll;

    // -----
    // Crea la interfaz gráfica de usuario
    public void init() {
        getContentPane().setLayout(null);

        lblMinutosTar = new JLabel("Minutos de tardanza");
        lblMinutosTar.setBounds(15, 15, 160, 23);
        getContentPane().add(lblMinutosTar);

        lblNumeroObs = new JLabel("Numero de observaciones");
        lblNumeroObs.setBounds(15, 39, 160, 23);
        getContentPane().add(lblNumeroObs);

        txtMinutosTar = new JTextField();
        txtMinutosTar.setBounds(175, 15, 120, 23);
        getContentPane().add(txtMinutosTar);

        txtNumeroObs = new JTextField();
        txtNumeroObs.setBounds(175, 39, 120, 23);
        getContentPane().add(txtNumeroObs);

        btnProcesar = new JButton("Procesar");
        btnProcesar.setBounds(365, 15, 100, 23);
        btnProcesar.addActionListener(this);
        getContentPane().add(btnProcesar);

        btnBorrar = new JButton("Borrar");
        btnBorrar.setBounds(365, 39, 100, 23);
        btnBorrar.addActionListener(this);
        getContentPane().add(btnBorrar);

        txtS = new JTextArea();
        txtS.setFont(new Font("monospaced", 0, 12));

        scpScroll = new JScrollPane(txtS);
```



```

        scpScroll.setBounds(15, 77, 450, 100);
        getContentPane().add(scpScroll);
    }
    // -----
    // Procesa eventos de tipo(ActionEvent)
    public void actionPerformed(ActionEvent e){
        if(e.getSource() == btnProcesar)
            procesar();
        if(e.getSource() == btnBorrar)
            borrar();
    }
    // -----
    // Método del botón Procesar
    void procesar(){
        // Declaración de variables locales
        int minutosTar, numeroObs, puntajePun, puntajeRen, puntajeTot;
        double bonificacion;

        // Entrada de datos
        minutosTar = getMinutos();
        numeroObs = getObservaciones();

        // Proceso de cálculo
        puntajePun = determinarPuntajePuntualidad(minutosTar);
        puntajeRen = determinarPuntajeRendimiento(numeroObs);
        puntajeTot = determinarPuntajeTotal(puntajePun, puntajeRen);
        bonificacion = determinarBonificacion(puntajeTot);

        // Salida de resultados
        txtS.setText("");
        imprimir("Puntaje por puntualidad : " + puntajePun);
        imprimir("Puntaje por rendimiento : " + puntajeRen);
        imprimir("Puntaje total : " + puntajeTot);
        imprimir("Bonificación : " + bonificacion);
    }
    // -----
    // Método del botón Borrar
    void borrar(){
        txtMinutosTar.setText("");
        txtNumeroObs.setText("");
        txtS.setText("");
        txtMinutosTar.requestFocus();
    }
    // -----
    // Lee y retorna los minutos de tardanza
    int getMinutos(){
        return Integer.parseInt(txtMinutosTar.getText());
    }
    // -----
    // Lee y retorna el número de observaciones
    int getObservaciones(){
        return Integer.parseInt(txtNumeroObs.getText());
    }
    // -----
    // Determina y retorna el puntaje por puntualidad
    int determinarPuntajePuntualidad(int minutosTar){
        if(minutosTar == 0)
            return 10;
        else if(minutosTar <= 2)
            return 8;
        else if(minutosTar <= 5)
            return 6;
        else if(minutosTar <= 9)
            return 4;
        else
            return 0;
    }
    // -----

```

```

// Determina y retorna el puntaje por rendimiento
int determinarPuntajeRendimiento(int numeroObs){
    if(numeroObs == 0)
        return 10;
    else if(numeroObs == 1)
        return 8;
    else if(numeroObs == 2)
        return 5;
    else if(numeroObs == 3)
        return 1;
    else
        return 0;
}
// -----
// Determina y retorna el puntaje total
int determinarPuntajeTotal(int puntajePun, int puntajeRen){
    return puntajePun + puntajeRen;
}
// -----
// Determina y retorna la bonificación
double determinarBonificacion(int puntajeTot){
    if(puntajeTot < 11)
        return 2.5*puntajeTot;
    else if(puntajeTot <= 13)
        return 5.0*puntajeTot;
    else if(puntajeTot <= 16)
        return 7.5*puntajeTot;
    else if(puntajeTot <= 19)
        return 10.0*puntajeTot;
    else
        return 12.5*puntajeTot;
}
// -----
// Imprime una cadena de texto incluyendo un salto de línea al final
void imprimir(String cad){
    txtS.append(cad + "\n");
}
}

```

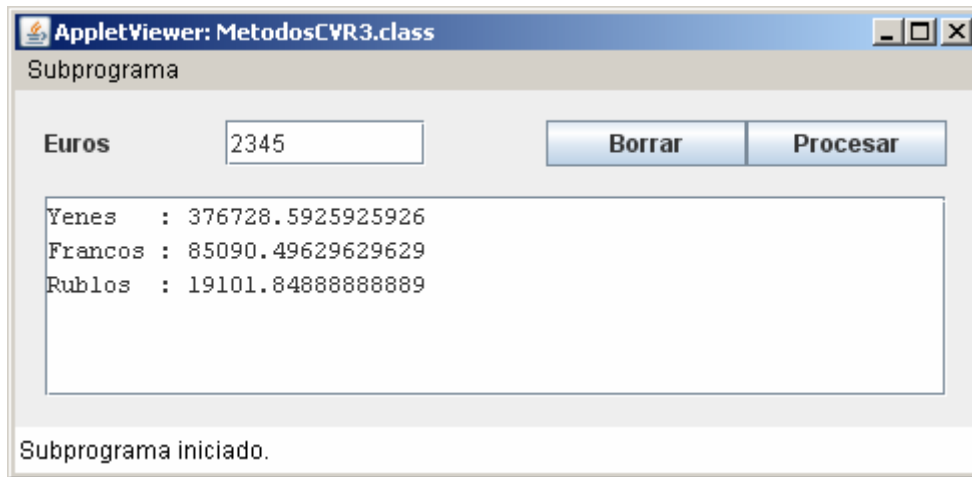
Problema 8.3

Diseñe un programa que permita convertir un monto de dinero dado en euros a sus equivalentes en yenes, francos y rublos. Considere que

1 dólar = 0.6750 euros
 1 dólar = 108.44 yenes
 1 dólar = 24.493 francos
 1 dólar = 5.4984 rublos

Declare todas las variables como locales y use métodos con valor de retorno para cada una de las entradas y para cada uno de los cálculos.

Programa



```
import java.awt.event.*;
import java.awt.*;
import javax.swing.*;

public class MetodosCVR3 extends JApplet implements ActionListener{
    // Declaración de variables globales
    JButton btnProcesar, btnBorrar;
    JTextArea txtS;
    JLabel lblEuros;
    JTextField txtEuros;
    JScrollPane scpScroll;

    // -----
    // Crea la interfaz gráfica de usuario
    public void init() {
        getContentPane().setLayout(null);

        lblEuros = new JLabel("Euros");
        lblEuros.setBounds(15, 15, 90, 23);
        getContentPane().add(lblEuros);

        txtEuros = new JTextField();
        txtEuros.setBounds(105, 15, 100, 23);
        getContentPane().add(txtEuros);

        btnProcesar = new JButton("Procesar");
        btnProcesar.setBounds(365, 15, 100, 23);
        btnProcesar.addActionListener(this);
        getContentPane().add(btnProcesar);

        btnBorrar = new JButton("Borrar");
        btnBorrar.setBounds(265, 15, 100, 23);
        btnBorrar.addActionListener(this);
        getContentPane().add(btnBorrar);

        txtS = new JTextArea();
        txtS.setFont(new Font("monospaced", 0, 12));

        scpScroll = new JScrollPane(txtS);
        scpScroll.setBounds(15, 53, 450, 100);
        getContentPane().add(scpScroll);
    }
    // -----
    // Procesa eventos de tipo(ActionEvent)
    public void actionPerformed( ActionEvent e ){
        if(e.getSource() == btnProcesar)
            procesar();
        if(e.getSource() == btnBorrar)
            borrar();
    }
}
```

```

// -----
// Método del botón Procesar
void procesar(){
    // Declaración de variables locales
    double meuros, myenes, mfrancos, mrublos;

    // Entrada de datos
    meuros = getEuros();

    // Proceso de cálculo
    myenes = convertirAYenes(meuros);
    mfrancos = convertirAFrancos(meuros);
    mrublos = convertirARublos(meuros);

    // Salida de resultados
    txtS.setText("");
    imprimir("Yenes : " + myenes);
    imprimir("Francos : " + mfrancos);
    imprimir("Rublos : " + mrublos);
}
// -----
// Método del botón Borrar
void borrar(){
    txtEuros.setText("");
    txtS.setText("");
    txtEuros.requestFocus();
}
// -----
// Lee y retorna la cantidad de euros
double getEuros(){
    return Double.parseDouble(txtEuros.getText());
}
// -----
// Convierte a yenes
double convertirAYenes(double xeuros){
    return (xeuros/0.6750)*108.44;
}
// -----
// Convierte a francos
double convertirAFrancos(double xeuros){
    return (xeuros/0.6750)*24.493;
}
// -----
// Convierte a rublos
double convertirARublos(double xeuros){
    return (xeuros/0.6750)*5.4984;
}
// -----
// Imprime una cadena de texto incluyendo un salto de línea al final
void imprimir(String cad){
    txtS.append(cad + "\n");
}
}

```

Problema 8.4

Diseñe un programa que determine el área lateral, el área total y el área de la base de un cilindro. Considere que:

$$areabase = \pi r^2$$

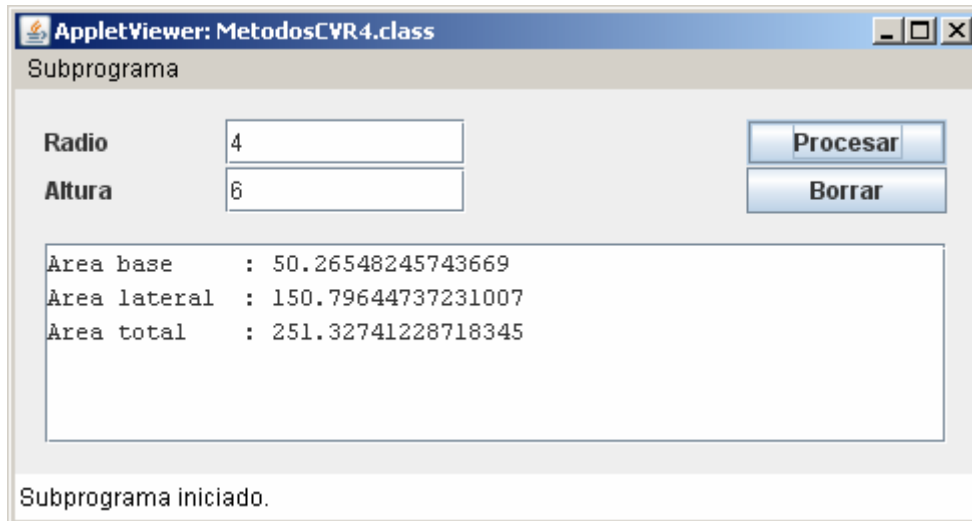
$$arealateral = 2\pi r h$$

$$areatotal = 2\pi r(r + h)$$

Donde r es el radio y h la altura.

Declare todas las variables como locales y use métodos con valor de retorno para cada una de las entradas y para cada uno de los cálculos.

Programa



```
import java.awt.event.*;
import java.awt.*;
import javax.swing.*;
```

```
public class MetodosCVR4 extends JApplet implements ActionListener{
    // Declaración de variables globales
    JButton btnProcesar, btnBorrar;
    JTextArea txtS;
    JLabel lblRadio, lblAltura;
    JTextField txtRadio, txtAltura;
    JScrollPane scpScroll;

    // -----
    // Crea la interfaz gráfica de usuario
    public void init() {
        getContentPane().setLayout(null);

        lblRadio = new JLabel("Radio");
        lblRadio.setBounds(15, 15, 90, 23);
        getContentPane().add(lblRadio);

        lblAltura = new JLabel("Altura");
        lblAltura.setBounds(15, 39, 90, 23);
        getContentPane().add(lblAltura);

        txtRadio = new JTextField();
        txtRadio.setBounds(105, 15, 120, 23);
        getContentPane().add(txtRadio);

        txtAltura = new JTextField();
        txtAltura.setBounds(105, 39, 120, 23);
        getContentPane().add(txtAltura);

        btnProcesar = new JButton("Procesar");
        btnProcesar.setBounds(365, 15, 100, 23);
        btnProcesar.addActionListener(this);
        getContentPane().add(btnProcesar);

        btnBorrar = new JButton("Borrar");
```

```

        btnBorrar.setBounds(365, 39, 100, 23);
        btnBorrar.addActionListener(this);
        getContentPane().add(btnBorrar);

        txtS = new JTextArea();
        txtS.setFont(new Font("monospaced", 0, 12));

        scpScroll = new JScrollPane(txtS);
        scpScroll.setBounds(15, 77, 450, 100);
        getContentPane().add(scpScroll);
    }
    // -----
    // Procesa eventos de tipo(ActionEvent)
    public void actionPerformed( ActionEvent e ){
        if(e.getSource() == btnProcesar)
            procesar();
        if(e.getSource() == btnBorrar)
            borrar();
    }
    // -----
    // Método del botón Procesar
    void procesar(){
        // Declaración de variables locales
        double radio, altura, areabas, arealat, areatot;

        // Entrada de datos
        radio = getRadio();
        altura = getAltura();

        // Proceso de cálculo
        areabas = calcularAreaBase(radio);
        arealat = calcularAreaLateral(radio, altura);
        areatot = calcularAreaTotal(radio, altura);

        // Salida de resultados
        txtS.setText("");
        imprimir("Area base      : " + areabas);
        imprimir("Area lateral   : " + arealat);
        imprimir("Area total     : " + areatot);
    }
    // -----
    // Método del botón Borrar
    void borrar(){
        txtRadio.setText("");
        txtAltura.setText("");
        txtS.setText("");
        txtRadio.requestFocus();
    }
    // -----
    // Lee y retorna el radio
    double getRadio(){
        return Double.parseDouble(txtRadio.getText());
    }
    // -----
    // Lee y retorna la altura
    double getAltura(){
        return Double.parseDouble(txtAltura.getText());
    }
    // -----
    // Calcula el área de la base
    double calcularAreaBase(double radio){
        return Math.PI*radio*radio;
    }
    // -----
    // Calcula el área lateral
    double calcularAreaLateral(double radio, double altura){
        return 2*Math.PI*radio*altura;
    }
}

```

```

// -----
// Calcula y retorna el área total
double calcularAreaTotal(double radio, double altura){
    return 2*Math.PI*radio*(radio+ altura);
}
// -----
// Imprime una línea de texto con un salto de línea al final
void imprimir( String cad ) {
    txtS.append(cad + "\n");
}
}

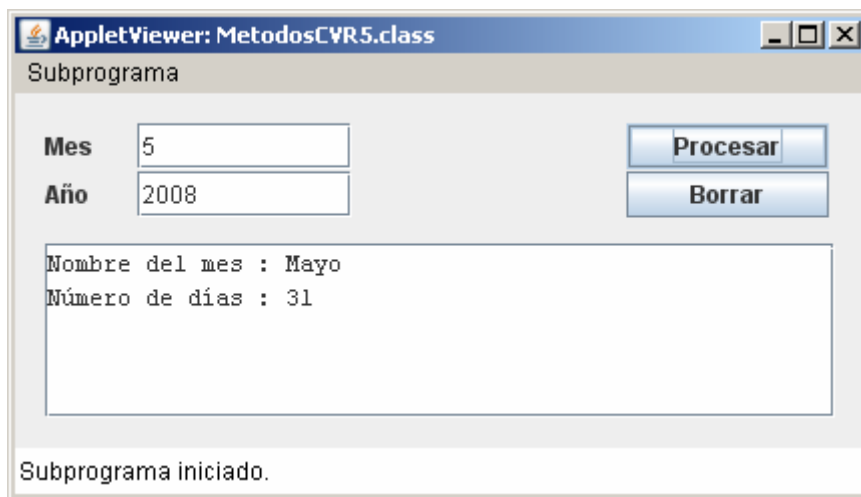
```

Problema 8.5

Diseñe un programa que determine la cantidad de días y el nombre de un mes conociendo los valores numéricos del mes y del año.

Declare todas las variables como locales y use métodos con valor de retorno

Programa



```

import java.awt.event.*;
import java.awt.*;
import javax.swing.*;

public class MetodosCVR5 extends JApplet implements ActionListener{
    // Declaración de variables globales
    JButton btnProcesar, btnBorrar;
    JTextArea txtS;
    JLabel lblAño, lblMes;
    JTextField txtAño, txtMes;
    JScrollPane scpScroll;

    // -----
    // Crea la interfaz gráfica de usuario GUI
    public void init() {
        getContentPane().setLayout(null);

        txtMes = new JTextField();
        txtMes.setBounds(61, 15, 107, 23);
        getContentPane().add(txtMes);

        txtAño = new JTextField();
        txtAño.setBounds(61, 39, 107, 23);
        getContentPane().add(txtAño);

        btnProcesar = new JButton("Procesar");

```

```

        btnProcesar.setBounds(305, 15, 101, 23);
        btnProcesar.addActionListener(this);
        getContentPane().add(btnProcesar);

        btnBorrar = new JButton("Borrar");
        btnBorrar.setBounds(305, 39, 101, 23);
        btnBorrar.addActionListener(this);
        getContentPane().add(btnBorrar);

        lblAño = new JLabel("Año");
        lblAño.setBounds(15, 39, 42, 23);
        getContentPane().add(lblAño);

        lblMes = new JLabel("Mes");
        lblMes.setBounds(15, 15, 42, 23);
        getContentPane().add(lblMes);

        txtS = new JTextArea();
        txtS.setFont(new Font("monospaced", 0, 12));

        scpScroll = new JScrollPane(txtS);
        scpScroll.setBounds(15, 75, 394, 87);
        getContentPane().add(scpScroll);
    }
    // -----
    // Procesa eventos de tipo ActionEvent
    public void actionPerformed( ActionEvent e ){
        if(e.getSource() == btnProcesar)
            procesar();
        if(e.getSource() == btnBorrar)
            borrar();
    }
    // -----
    // Método del botón Procesar
    void procesar(){

        // Declaración de variables locales
        int mes, año, dias;
        String nombre;

        // Entrada de datos
        mes = Integer.parseInt(txtMes.getText());
        año = Integer.parseInt(txtAño.getText());

        // Proceso
        nombre = determinaNombreMes(mes);
        dias = determinaNumeroDiasMes(mes, año);

        // Salida de resultados
        txtS.setText("");
        imprimir("Nombre del mes : " + nombre);
        imprimir("Número de días : " + dias);
    }
    // -----
    // Método del botón Borrar
    void borrar(){
        txtS.setText("");
        txtMes.setText("");
        txtAño.setText("");
        txtMes.requestFocus();
    }
    // -----
    // Determina el nombre del mes
    String determinaNombreMes(int mes){
        String nombre;

        switch(mes){
            case 1 :

```



```

        nombre = "Enero";
        break;
    case 2 :
        nombre = "Febrero";
        break;
    case 3 :
        nombre = "Marzo";
        break;
    case 4 :
        nombre = "Abril";
        break;
    case 5 :
        nombre = "Mayo";
        break;
    case 6 :
        nombre = "Junio";
        break;
    case 7 :
        nombre = "Julio";
        break;
    case 8 :
        nombre = "Agosto";
        break;
    case 9 :
        nombre = "Setiembre";
        break;
    case 10:
        nombre = "Octubre";
        break;
    case 11:
        nombre = "Noviembre";
        break;
    default:
        nombre = "Diciembre";
    }

    return nombre;
}
// -----
// Determina el número de días del mes
int determinaNumeroDiasMes(int mes, int año){
    int dias;

    switch(mes){
        case 1: case 3: case 5: case 7: case 8: case 10: case 12:
            dias = 31;
            break;
        case 4: case 6: case 9: case 11:
            dias = 30;
            break;
        default:
            if((año%4 == 0) && ((año%100 != 0) || (año%400 == 0)))
                dias = 29;
            else
                dias = 28;
    }

    return dias;
}
// -----
// Imprime una cadena de texto incluyendo un salto de línea al final
void imprimir(String cad){
    txtS.append(cad + "\n");
}
}

```

Problemas propuestos

Actividad

- Una tienda vende tres tipos de productos a los precios unitarios dados en la siguiente tabla:

Producto	Precio
P1	S/. 15.0
P2	S/. 17.5
P3	S/. 20.0

Como oferta la tienda ofrece un regalo de acuerdo a la siguiente tabla:

Unidades adquiridas	Regalo
1 a 25	un lapicero
26 a 50	un cuaderno
Más de 50	una agenda

Diseñe un programa que determine el importe a pagar y el regalo para un cliente de la tienda. Declare todas las variables como locales y use métodos con valor de retorno para cada una de las entrada y para cada uno de los cálculos

- En un parque de diversiones un juego consiste en efectuar 20 lanzamientos de una pelota a uno de cuatro arcos de diferentes tamaños. Por cada gol anotado se recibe cierta cantidad de puntos de acuerdo a la siguiente tabla:

Arco	Puntos por gol
Pequeño	6
Mediano	4
Grande	3
Extra Grande	2

Luego en base al puntaje total obtenido se efectúa un obsequio de acuerdo a la siguiente tabla:

Puntaje total	Obsequio
0 a 9	Ninguno
10 a 39	Pelucho
Más de 39	Reloj

Diseñe un programa que determine el puntaje total obtenido por un jugador y el premio que le corresponde.

Declare todas las variables como locales y use métodos con valor de retorno para cada una de las entrada y para cada uno de los cálculos.

Autoevaluación

1. Resuelva los problemas 9, 10, 11, 12, 16, 18, 19 y 20 de la Autoevaluación de la semana 8 considerando todas las variables como locales y usando métodos con valor de retorno para cada una de las entradas y para cada uno de los cálculos.
2. Determine que se imprime al ejecutar el método **procesar** correspondiente al botón **Procesar** de un programa:

```
void procesar(){
    int a, b, c, d;
    a = metodo1();
    b = metodo2();
    c = metodo3(a, b);
    d = metodo3(12, 7);
    txtS.append(a + "\n");
    txtS.append(b + "\n");
    txtS.append(c + "\n");
    txtS.append(d + "\n");
}

int metodo1(){
    return 20;
}

int metodo2(){
    return 30;
}

int metodo3(int p, int q){
    int r = p+q;
    if(r%2 == 0)
        return 1;
    else
        return 2;
}
```

3. Determine que se imprime al ejecutar el método **procesar** correspondiente al botón **Procesar** de un programa:

```
void procesar(){
    int a, b, c, d;
    a = metodo1(50);
    b = metodo2(30);
    c = metodo3(a+5, b+5);
    d = metodo4(c);
    txtS.append(a + "\n");
    txtS.append(b + "\n");
    txtS.append(c + "\n");
    txtS.append(d + "\n");
}

int metodo1(int p){
    return p-20;
}

int metodo2(int p){
```

```
        return p+20;
    }

    int metodo3(int p, int q){
        return p+q+1;
    }

    int metodo4(int c){
        int s;
        s = metodo3(c, c+1) + metodo1(10);
        return s;
    }
```

Para recordar

- Un método con valor de retorno es un método que puede recibir datos de entrada a través de variables locales al método conocidas como parámetros y que retorna un valor al punto donde es invocado.

UNIDAD DE
APRENDIZAJE

5

SEMANA

12

CONTADORES Y ACUMULADORES

LOGRO DE LA UNIDAD DE APRENDIZAJE

Al finalizar la unidad, los alumnos, utilizando variables globales y locales y los demás conceptos aprendidos en el curso, diseñarán programas que resuelvan problemas de conteo y acumulación,

CONTENIDO

1. Operadores de incremento y decremento
2. Operadores de asignación compleja
3. Contadores y acumuladores.

ACTIVIDADES

Los alumnos desarrollan programas usando contadores y acumuladores.

2. OPERADORES DE INCREMENTO Y DECREMENTO

Son operadores que permiten incrementar o decrementar en una unidad el valor de una variable numérica.

Operador	Uso	Equivalencia
++	a++;	a = a + 1;
--	a--;	a = a - 1;

Ejemplo 9.1

```
// Incrementa en uno el valor de x (Forma 1)
x = x + 1;

// Incrementa en uno el valor de x (Forma 2)
x++;

// Decrementa en 1 el valor de la variable z (Forma 1)
z = z - 1;

// Decrementa en 1 el valor de la variable z (Forma 2)
z--;
```

3. OPERADORES DE ASIGNACION COMPLEJA

Son operadores que permiten asignar a una variable el valor de la variable mas, menos, por o entre el valor de otra variable.

Operador	Ejemplo	Equivalencia
+=	a += b;	a = a + b;
-=	a -= b;	a = a - b;
*=	a *= b;	a = a * b;
/=	a /= b;	a = a / b;

Ejemplo 9.2

```
// Incrementa en 2 el valor de la variable z (Forma 1)
z = z + 2;

// Incrementa en 2 el valor de la variable z (Forma 2)
z += 2;

// Decrementa en 5 el valor de la variable m (Forma 1)
m = m - 5;

// Decrementa en 5 el valor de la variable m (Forma 2)
m -= 5;
```

4. CONTADORES

Un **contador** es una variable que se utiliza para contar el número de ocurrencias de un suceso o el número de veces que se cumple una determinada condición.

Por ejemplo, se necesita un **contador** para determinar:

- El número de veces que se hizo clic en un botón.
- El número de notas ingresadas.
- El número de notas aprobatorias.
- El número de notas desaprobatorias.
- Etc.

Una instrucción de conteo tiene la siguiente forma:

```
contador = contador +1;
```

Que puede escribirse también como:

```
contador++;
```

Ejemplo 9.3

```
// Incrementa el número de alumnos aprobados de una sección
aprobados++;

// Incrementa el número de ventas efectuadas en un día
numeroVentasDia++;
```

5. ACUMULADORES

Un **acumulador** es una variable que se utiliza para acumular o totalizar cantidades de una misma especie.

Por ejemplo, se necesita un **acumulador** para determinar:

- El sueldo total de los empleados de una empresa.
- La edad total de un conjunto de personas.
- La suma de las notas de un alumno.
- Etc.

Una instrucción de acumulación tiene la siguiente forma:

```
acumulador = acumulador + cantidad;
```

Qué puede escribirse también como:

```
acumulador += cantidad;
```

Ejemplo 9.3

```
// Incrementa el monto total vendido
montoTotalVendido += montoVenta;

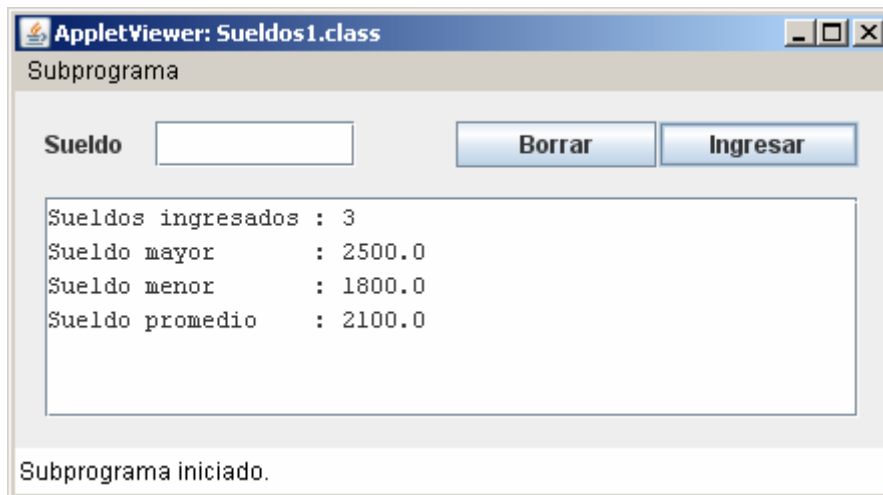
// Incrementa el sueldo total de los empleados de una empresa
sueldoTotalEmpresa += sueldoEmpleado;
```

6. PROBLEMAS RESUELTOS

Problema 9.1

Diseñe un programa que permita ingresar, mediante un botón y de uno en uno, los sueldos de los empleados de una empresa y muestre luego de cada ingreso la cantidad de sueldos ingresados, el mayor y el menor sueldo ingresado y, el sueldo promedio.

Programa 1



```
import java.awt.event.*;
import java.awt.*;
import javax.swing.*;
```

```
public class Sueldos1 extends JApplet implements ActionListener{
    // Declaración de variables globales para la GUI
    JButton btnIngresar, btnBorrar;
    JTextArea txtS;
    JLabel lblSueldo;
    JTextField txtSueldo;
    JScrollPane scpScroll;

    // Declaración de variables globales para el algoritmo
    int contaSuel=0;
    double suelMay, suelMen, suelTot=0, suelProm;

    // -----
    // Crea la interfaz gráfica de usuario
    public void init() {
        getContentPane().setLayout(null);

        txtSueldo = new JTextField();
        txtSueldo.setBounds(70, 15, 100, 23);
        getContentPane().add(txtSueldo);

        btnIngresar = new JButton("Ingresar");
        btnIngresar.setBounds(321, 15, 100, 23);
        btnIngresar.addActionListener(this);
        getContentPane().add(btnIngresar);

        btnBorrar = new JButton("Borrar");
        btnBorrar.setBounds(220, 15, 100, 23);
        btnBorrar.addActionListener(this);
        getContentPane().add(btnBorrar);
    }
}
```



```

        lblSueldo = new JLabel("Sueldo");
        lblSueldo.setBounds(15, 15, 90, 23);
        getContentPane().add(lblSueldo);

        txtS = new JTextArea();
        txtS.setFont(new Font("monospaced", 0, 12));

        scpScroll = new JScrollPane(txtS);
        scpScroll.setBounds(15, 53, 406, 110);
        getContentPane().add(scpScroll);
    }
    // -----
    // Procesa eventos de tipo ActionEvent
    public void actionPerformed( ActionEvent e ){
        if(e.getSource() == btnIngresar)
            procesar();
        if(e.getSource() == btnBorrar)
            borrar();
    }
    // -----
    // Método del botón Borrar
    void borrar(){
        txtSueldo.setText("");
        txtSueldo.requestFocus();
    }
    // -----
    // Método del botón Procesar
    void procesar(){
        double sueldo;
        sueldo = getSueldo();
        borrar();
        registrarEmpleado(sueldo);
        mostrarReporte();
    }
    // -----
    // Lee y retorna el sueldo
    double getSueldo(){
        return Double.parseDouble(txtSueldo.getText());
    }
    // -----
    // Registra el sueldo de un empleado
    void registrarEmpleado(double sueldo){

        // Cuenta el sueldo ingresado
        contaSuel++;

        // Acumula el sueldo ingresado
        suelTot += sueldo;

        // Promedia el sueldo
        suelProm = suelTot/contaSuel;

        // Determina el mayor y el menor sueldo ingresados
        if( contaSuel == 1 ){
            suelMay = sueldo;
            suelMen = sueldo;
        }
        else{
            if( sueldo > suelMay )
                suelMay = sueldo;
            if( sueldo < suelMen )
                suelMen = sueldo;
        }
    }
    // -----
    // Imprime una cadena con un salto de línea
    void imprimir(String cad){
        txtS.append(cad + "\n");
    }

```

```

    }
    // -----
    // Muestra el reporte
    void mostrarReporte(){
        txtS.setText("");
        imprimir("Sueldos ingresados : " + contaSuel);
        imprimir("Sueldo mayor      : " + suelMay);
        imprimir("Sueldo menor      : " + suelMen);
        imprimir("Sueldo promedio   : " + suelProm);
    }
}

```

Programa 2

```

import java.awt.event.*;
import java.awt.*;
import javax.swing.*;

public class Sueldos2 extends JApplet implements ActionListener{
    // Declaración de variables globales para la GUI
    JButton btnIngresar, btnBorrar;
    JTextArea txtS;
    JLabel lblSueldo;
    JTextField txtSueldo;
    JScrollPane scpScroll;

    // Declaración de variables globales para el algoritmo
    int contaSuel=0;
    double suelMay=0, suelMen=Double.MAX_VALUE, suelTot=0, suelProm;

    // -----
    // Crea la interfaz gráfica de usuario
    public void init() {
        getContentPane().setLayout(null);

        txtSueldo = new JTextField();
        txtSueldo.setBounds(70, 15, 100, 23);
        getContentPane().add(txtSueldo);

        btnIngresar = new JButton("Ingresar");
        btnIngresar.setBounds(321, 15, 100, 23);
        btnIngresar.addActionListener(this);
        getContentPane().add(btnIngresar);

        btnBorrar = new JButton("Borrar");
        btnBorrar.setBounds(220, 15, 100, 23);
        btnBorrar.addActionListener(this);
        getContentPane().add(btnBorrar);

        lblSueldo = new JLabel("Sueldo");
        lblSueldo.setBounds(15, 15, 90, 23);
        getContentPane().add(lblSueldo);

        txtS = new JTextArea();
        txtS.setFont(new Font("monospaced", 0, 12));

        scpScroll = new JScrollPane(txtS);
        scpScroll.setBounds(15, 53, 406, 110);
        getContentPane().add(scpScroll);
    }
    // -----
    // Procesa eventos de tipo ActionEvent
    public void actionPerformed( ActionEvent e ){
        if(e.getSource() == btnIngresar)
            procesar();
        if(e.getSource() == btnBorrar)
            borrar();
    }
}

```

```

// -----
// Método del botón Borrar
void borrar(){
    txtSueldo.setText("");
    txtSueldo.requestFocus();
}
// -----
// Método del botón Procesar
void procesar(){
    double sueldo;
    sueldo = getSueldo();
    borrar();
    registrarEmpleado(sueldo);
    mostrarReporte();
}
// -----
// Lee y retorna el sueldo
double getSueldo(){
    return Double.parseDouble(txtSueldo.getText());
}
// -----
// Registra el sueldo de un empleado
void registrarEmpleado(double sueldo){

    // Cuenta el sueldo ingresado
    contaSuel++;

    // Acumula el sueldo ingresado
    suelTot += sueldo;

    // Promedia el sueldo
    suelProm = suelTot/contaSuel;

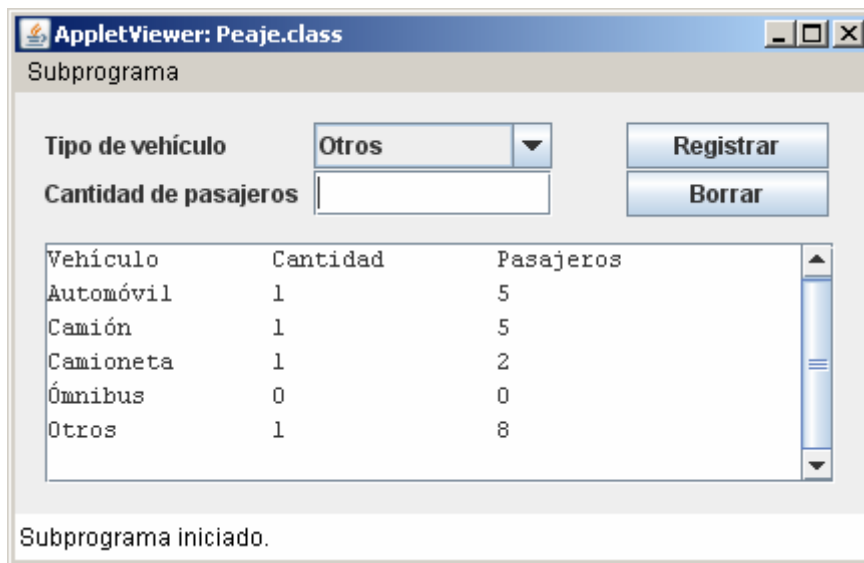
    // Determina el mayor y el menor sueldo ingresados
    if( sueldo > suelMay )
        suelMay = sueldo;
    if( sueldo < suelMen )
        suelMen = sueldo;
}
// -----
// Imprime una cadena con un salto de línea
void imprimir(String cad){
    txtS.append(cad + "\n");
}
// -----
// Muestra el reporte
void mostrarReporte(){
    txtS.setText("");
    imprimir("Sueldos ingresados : " + contaSuel);
    imprimir("Sueldo mayor      : " + suelMay);
    imprimir("Sueldo menor       : " + suelMen);
    imprimir("Sueldo promedio    : " + suelProm);
}
}

```

Problema 9.2

En un peaje se desea saber cuantos vehículos de cada tipo pasaron por el peaje y la cantidad total de pasajeros por tipo de vehículo. Los tipos de vehículos a considerar son: automóvil, camión, camioneta, omnibus y otros.

Programa



```
import java.awt.event.*;
import java.awt.*;
import javax.swing.*;

public class Peaje extends JApplet implements ActionListener{
    // Declaración de variables globales para la GUI
    JButton btnRegistrar, btnBorrar;
    JTextArea txtS;
    JComboBox cboVehiculo;
    JLabel lblPasajeros, lblTipo;
    JTextField txtPasajeros;
    JScrollPane scpScroll;

    // Declaración de variables globales para el algoritmo
    int contaveh0=0, contaveh1=0, contaveh2=0, contaveh3=0, contaveh4=0;
    int totpasaj0=0, totpasaj1=0, totpasaj2=0, totpasaj3=0, totpasaj4=0;

    // -----
    // Crea la interfaz gráfica de usuario GUI
    public void init() {
        getContentPane().setLayout(null);

        lblPasajeros = new JLabel("Cantidad de pasajeros");
        lblPasajeros.setBounds(15, 39, 131, 23);
        getContentPane().add(lblPasajeros);

        lblTipo = new JLabel("Tipo de vehículo");
        lblTipo.setBounds(15, 15, 105, 23);
        getContentPane().add(lblTipo);

        txtPasajeros = new JTextField();
        txtPasajeros.setBounds(149, 39, 119, 23);
        getContentPane().add(txtPasajeros);

        cboVehiculo = new JComboBox();
        cboVehiculo.setBounds(149, 15, 119, 23);
        cboVehiculo.addItem("Automovil");
        cboVehiculo.addItem("Camión");
        cboVehiculo.addItem("Camioneta");
        cboVehiculo.addItem("Ómnibus");
        cboVehiculo.addItem("Otros");
        getContentPane().add(cboVehiculo);
    }
}
```

```

        btnRegistrar = new JButton("Registrar");
        btnRegistrar.setBounds(305, 15, 101, 23);
        btnRegistrar.addActionListener(this);
        getContentPane().add(btnRegistrar);

        btnBorrar = new JButton("Borrar");
        btnBorrar.setBounds(305, 39, 101, 23);
        btnBorrar.addActionListener(this);
        getContentPane().add(btnBorrar);

        txtS = new JTextArea();
        txtS.setFont(new Font("monospaced", 0, 12));

        scpScroll = new JScrollPane(txtS);
        scpScroll.setBounds(15, 75, 394, 120);
        getContentPane().add(scpScroll);
    }
    // -----
    // Procesa eventos de tipo(ActionEvent)
    public void actionPerformed(ActionEvent e){
        if(e.getSource() == btnRegistrar)
            procesar();
        if(e.getSource() == btnBorrar)
            borrar();
    }
    // -----
    // Método para el botón Borrar
    void borrar(){
        txtPasajeros.setText("");
        txtPasajeros.requestFocus();
    }
    // -----
    // Método para el botón Procesar
    void procesar(){
        int tipoveh, cantpasaj;

        tipoveh = getVehiculo();
        cantpasaj = getPasajeros();
        borrar();
        registrarVehiculo(tipoveh, cantpasaj);
        mostrarReporte();
    }
    // -----
    // Lee y retorna el tipo de vehículo
    int getVehiculo(){
        return cboVehiculo.getSelectedIndex();
    }
    // -----
    // Lee y retorna la cantidad de pasajeros
    int getPasajeros(){
        return Integer.parseInt(txtPasajeros.getText());
    }
    // -----
    // Registra un vehículo
    void registrarVehiculo(int tipoveh, int cantpasaj){
        switch(tipoveh){
            case 0:
                contaveh0++;
                totpasaj0 += cantpasaj;
                break;
            case 1:
                contaveh1++;
                totpasaj1 += cantpasaj;
                break;
            case 2:
                contaveh2++;
                totpasaj2 += cantpasaj;
                break;
        }
    }

```

```

        case 3:
            contaveh3++;
            totpasaj3 += cantpasaj;
            break;
        default:
            contaveh4++;
            totpasaj4 += cantpasaj;
    }
}
// -----
// Imprime una cadena con un salto de línea
void imprimir(String cad){
    txtS.append(cad + "\n");
}
// -----
// Muestra el reporte
void mostrarReporte(){
    txtS.setText("Vehículo\tCantidad\tPasajeros\n");
    imprimir("Automóvil" + "\t" + contaveh0 + "\t\t" + totpasaj0);
    imprimir("Camión    " + "\t" + contaveh1 + "\t\t" + totpasaj1);
    imprimir("Camioneta" + "\t" + contaveh2 + "\t\t" + totpasaj2);
    imprimir("Ómnibus   " + "\t" + contaveh3 + "\t\t" + totpasaj3);
    imprimir("Otros     " + "\t" + contaveh4 + "\t\t" + totpasaj4);
}
}

```

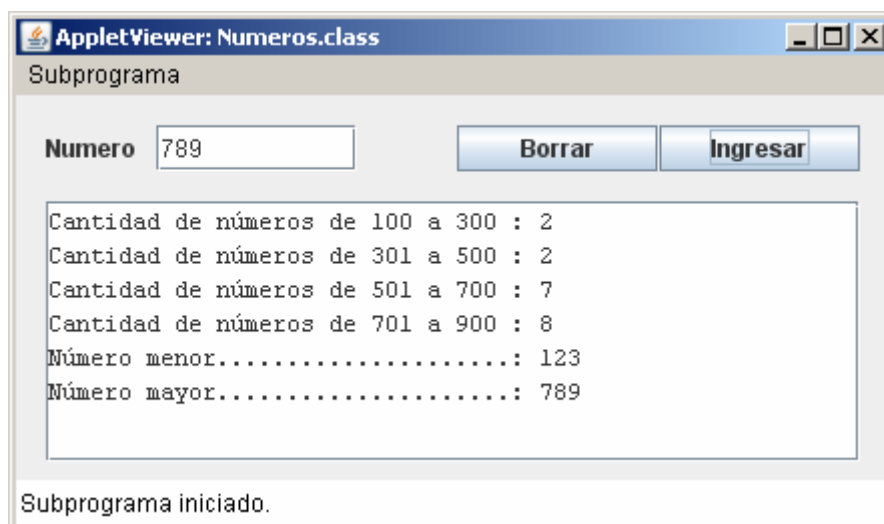
Problema 9.3

Diseñe un programa que permita ingresar, mediante un botón y de uno en uno, números enteros mayores o iguales que 100 y pero menores o iguales que 900 y determine:

Cuántos números pertenecen a cada uno de los siguientes intervalos: 100 a 300, 301 a 500, 501 a 700 y 701 a 900.

El mayor y el menor número ingresados.

Programa



```

import java.awt.event.*;
import java.awt.*;
import javax.swing.*;

public class Numeros extends JApplet implements ActionListener{
    // Declaración de variables globales para la GUI

```

```

JButton btnIngresar, btnBorrar;
JTextArea txtS;
JLabel lblNumero;
JTextField txtNumero;
JScrollPane scpScroll;

// Declaración de variables globales para el algoritmo
int c1=0, c2=0, c3=0, c4=0;
int mayor=99, menor=901;

// Crea la interfaz gráfica de usuario
public void init() {
    getContentPane().setLayout(null);

    lblNumero = new JLabel("Numero");
    lblNumero.setBounds(15, 15, 90, 23);
    getContentPane().add(lblNumero);

    txtNumero = new JTextField();
    txtNumero.setBounds(70, 15, 100, 23);
    getContentPane().add(txtNumero);

    btnIngresar = new JButton("Ingresar");
    btnIngresar.setBounds(321, 15, 100, 23);
    btnIngresar.addActionListener(this);
    getContentPane().add(btnIngresar);

    btnBorrar = new JButton("Borrar");
    btnBorrar.setBounds(220, 15, 100, 23);
    btnBorrar.addActionListener(this);
    getContentPane().add(btnBorrar);

    txtS = new JTextArea();
    txtS.setFont(new Font("monospaced", 0, 12));

    scpScroll = new JScrollPane(txtS);
    scpScroll.setBounds(15, 53, 406, 130);
    getContentPane().add(scpScroll);
}
// -----
// Procesa eventos de tipo(ActionEvent)
public void actionPerformed(ActionEvent e){
    if( e.getSource() == btnIngresar )
        ingresar();
    if( e.getSource() == btnBorrar )
        borrar();
}
// -----
// Método del botón Borrar
void borrar(){
    txtNumero.setText("");
    txtNumero.requestFocus();
}
// -----
// Método del botón Ingresar
void ingresar(){
    int numero;
    numero = getNumero();
    if( numero >= 100 && numero <= 900 ){
        contarNumero(numero);
        determinarMenor(numero);
        determinarMayor(numero);
        mostrarReporte();
    }
    else
        JOptionPane.showMessageDialog(this,
            "Ingrese un número de 100 a 900");
}

```

```

// -----
// Lee y retorna el número
int getNumero(){
    return Integer.parseInt(txtNumero.getText());
}
// -----
// Limpia la caja de entrada
void limpiarEntrada(){
    txtNumero.setText("");
    txtNumero.requestFocus();
}
// -----
// Cuenta el número ingresado
void contarNumero(int numero){
    if( numero <= 300 )
        c1++;
    else if( numero <= 500 )
        c2++;
    else if( numero <= 700 )
        c3++;
    else
        c4++;
}
// -----
// Determina el menor número ingresado
void determinarMenor(int numero){
    if( numero < menor )
        menor = numero;
}
// -----
// Determina el mayor número ingresado
void determinarMayor(int numero){
    if( numero > mayor )
        mayor = numero;
}
// -----
// Imprime una cadena con un salto de línea
void imprimir(String cad){
    txtS.append(cad + "\n");
}
// -----
// Muestra el reporte
void mostrarReporte(){
    txtS.setText("");
    imprimir("Cantidad de números de 100 a 300 : " + c1);
    imprimir("Cantidad de números de 301 a 500 : " + c2);
    imprimir("Cantidad de números de 501 a 700 : " + c3);
    imprimir("Cantidad de números de 701 a 900 : " + c4);
    imprimir("Número menor.....: " + menor);
    imprimir("Número mayor.....: " + mayor);
}
}

```

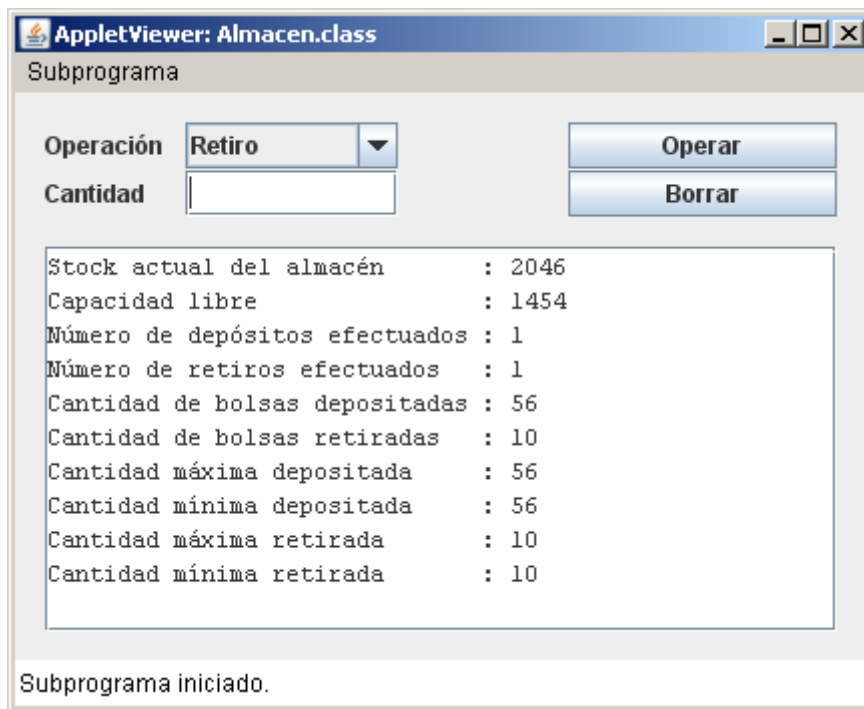
Problema 9.4

Diseñe un programa para simular el movimiento de un almacén de bolsas de cemento. Al comenzar la ejecución del programa, inicializar el stock de bolsas de cemento en 2000 bolsas y la capacidad del almacén en 3500 bolsas. Luego, el programa permitirá efectuar operaciones de depósito y de retiro, mostrando en todo momento:

- El stock actual del almacén.
- La capacidad libre del almacén.
- El número de depósitos y de retiros efectuados.
- La cantidad total de bolsas depositadas y retiradas.

- La cantidad máxima de bolsas retiradas y depositadas.
- La cantidad mínima de bolsas retiradas y depositadas.

Programa



```
import java.awt.event.*;
import java.awt.*;
import javax.swing.*;

public class Almacen extends JApplet implements ActionListener{
    // Declaración de variables globales para la GUI
    JButton btnBorrar, btnOperar;
    JTextArea txtS;
    JLabel lblCantidad, lblOperacion;
    JComboBox cboOperacion;
    JTextField txtCantidad;
    JScrollPane scpScroll;

    // Declaración de variables para el algoritmo
    int stock = 2000, capacidad = 3500;
    int contaret, contadep;
    int bolret, boldep;
    int maxret, minret, maxdep, mindep;

    // -----
    // Crea la interfaz gráfica de usuario GUI
    public void init() {
        getContentPane().setLayout(null);

        lblCantidad = new JLabel("Cantidad");
        lblCantidad.setBounds(15, 39, 60, 23);
        getContentPane().add(lblCantidad);

        lblOperacion = new JLabel("Operación");
        lblOperacion.setBounds(15, 15, 62, 23);
        getContentPane().add(lblOperacion);

        txtCantidad = new JTextField();
        txtCantidad.setBounds(85, 39, 106, 23);
```

```

        getContentPane().add(txtCantidad);

        cboOperacion = new JComboBox();
        cboOperacion.setBounds(85, 15, 106, 23);
        cboOperacion.addItem("Depósito");
        cboOperacion.addItem("Retiro");
        getContentPane().add(cboOperacion);

        btnOperar = new JButton("Operar");
        btnOperar.setBounds(276, 15, 134, 23);
        btnOperar.addActionListener(this);
        getContentPane().add(btnOperar);

        btnBorrar = new JButton("Borrar");
        btnBorrar.setBounds(276, 39, 134, 23);
        btnBorrar.addActionListener(this);
        getContentPane().add(btnBorrar);

        txtS = new JTextArea();
        txtS.setFont(new Font("monospaced", 0, 12));

        scpScroll = new JScrollPane(txtS);
        scpScroll.setBounds(15, 77, 395, 192);
        getContentPane().add(scpScroll);
        txtS.setEditable(false);

        mostrarReporte();
    }
    // -----
    // Procesa eventos de tipo ActionEvent
    public void actionPerformed( ActionEvent e ){
        if( e.getSource() == btnOperar )
            operar();
        if( e.getSource() == btnBorrar )
            borrar();
    }
    // -----
    // Método del botón Borrar
    void borrar(){
        txtCantidad.setText("");
        txtCantidad.requestFocus();
    }
    // -----
    // Lee y retorna el tipo de operación
    int getOperacion(){
        return cboOperacion.getSelectedIndex();
    }
    // -----
    // Lee y retorna la cantidad de bolsas
    int getCantidad(){
        return Integer.parseInt(txtCantidad.getText());
    }
    // -----
    // Método del botón Operar
    void operar(){
        int operacion, cantidad, caplibre;

        operacion = getOperacion();
        cantidad = getCantidad();
        limpiarEntrada();
        if( operacion == 0 )
            depositar(cantidad);
        else
            retirar(cantidad);
        mostrarReporte();
    }
    // -----
    // Retira una cantidad de bolsas del almacén

```

```

void depositar(int cantidad){
    if( cantidad <= capacidad-stock ){
        stock += cantidad;
        contadep++;
        boldep += cantidad;
        if( contadep == 1 ){
            maxdep = cantidad;
            mindep = cantidad;
        }
        else{
            if( cantidad > maxdep )
                maxdep = cantidad;
            if( cantidad < mindep )
                mindep = cantidad;
        }
    }
    else
        JOptionPane.showMessageDialog(this,"Espacio libre insuficiente");
}
// -----
// Deposita una cantidad de bolsas al almacén
void retirar(int cantidad){
    if( cantidad <= stock ){
        stock -= cantidad;
        contaret++;
        bolret += cantidad;
        if( contaret == 1 ){
            maxret = cantidad;
            minret = cantidad;
        }
        else{
            if( cantidad > maxret )
                maxret = cantidad;
            if( cantidad < minret )
                minret = cantidad;
        }
    }
    else
        JOptionPane.showMessageDialog(this,"Stock insuficiente");
}
// -----
// Muestra el reporte
void mostrarReporte(){
    txtS.setText("");
    imprimir("Stock actual del almacén          : " + stock);
    imprimir("Capacidad libre                    : " + (capacidad-stock));
    imprimir("Número de depósitos efectuados : " + contadep);
    imprimir("Número de retiros efectuados   : " + contaret);
    imprimir("Cantidad de bolsas depositadas : " + boldep);
    imprimir("Cantidad de bolsas retiradas   : " + bolret);
    imprimir("Cantidad máxima depositada     : " + maxdep);
    imprimir("Cantidad mínima depositada     : " + mindep);
    imprimir("Cantidad máxima retirada       : " + maxret);
    imprimir("Cantidad mínima retirada       : " + minret);
}
// -----
// Imprime una cadena con un salto de línea
void imprimir(String cad){
    txtS.append(cad + "\n");
}
// -----
// Limpia la caja de texto
void limpiarEntrada(){
    txtCantidad.setText("");
    txtCantidad.requestFocus();
}
}

```

Problema 9.5

Una empresa desarrolladora de software ha puesto a la venta licencias de su programa de edición de video Video Edit 2.0 a los siguientes costos unitarios:

Licencia	Costo
Cobre	\$ 510
Bronze	\$ 1500
Silver	\$ 3100
Gold	\$ 4500

Diseñe un programa que permita ingresar por cada venta el tipo de licencia y el número de licencias y muestre luego de cada venta:

- El importe a pagar para la venta efectuada.
- El importe total recaudado de cada tipo de licencia.
- El número de licencias vendidas de cada tipo de licencia.
- El número de ventas efectuadas de cada tipo de licencia.

Programa

AppletViewer: Licencias.class

Subprograma

Tipo de licencia: Gold

Numero de licencias: 2

Procesar

Borrar

Venta Actual

Tipo de licencia.....: Gold

Número de licencias.....: 2

Importe a pagar.....: 9000.0

Reporte Histórico

Número de licencias vendidas

Por licencias Cobre....: 4

Por licencias Bronze...: 6

Por licencias Silver...: 0

Por licencias Gold.....: 10

Número de ventas efectuadas

Por licencias Cobre....: 1

Por licencias Bronze...: 2

Por licencias Silver...: 0

Por licencias Gold.....: 5

Subprograma iniciado.

```
import java.awt.event.*;
import java.awt.*;
```

```

import javax.swing.*;

public class Licencias extends JApplet implements ActionListener{
    // Declaración de variables globales para la GUI
    JButton btnProcesar, btnBorrar;
    JTextArea txtS1, txtS2;
    JScrollPane scpScroll1, scpScroll2;
    JLabel lblTipoLicencia, lblNumeroLicencias, lblActual, lblHistorico;
    JTextField txtNumeroLicencias;
    JComboBox cboTipo;

    // Declaración de variables globales para el algoritmo
    double imptot1=0, imptot2=0, imptot3=0, imptot4=0;
    int nlic1=0, nlic2=0, nlic3=0, nlic4=0;
    int nven1=0, nven2=0, nven3=0, nven4=0;
    // -----
    // Crea la interfaz gráfica de usuario
    public void init() {
        setLayout(null);
        setBackground(new Color(214,211,206));

        cboTipo = new JComboBox();
        cboTipo.setBounds(155, 15, 170, 23);
        cboTipo.addItem("Cobre");
        cboTipo.addItem("Bronze");
        cboTipo.addItem("Silver");
        cboTipo.addItem("Gold");
        add(cboTipo);

        txtNumeroLicencias = new JTextField();
        txtNumeroLicencias.setBounds(155, 39, 170, 23);
        add(txtNumeroLicencias);

        btnProcesar = new JButton("Procesar");
        btnProcesar.setBounds(356, 15, 100, 23);
        btnProcesar.addActionListener(this);
        add(btnProcesar);

        btnBorrar = new JButton("Borrar");
        btnBorrar.setBounds(356, 39, 100, 23);
        btnBorrar.addActionListener(this);
        add(btnBorrar);

        lblTipoLicencia = new JLabel("Tipo de licencia");
        lblTipoLicencia.setBounds(15, 15, 140, 23);
        add(lblTipoLicencia);

        lblNumeroLicencias = new JLabel("Numero de licencias");
        lblNumeroLicencias.setBounds(15, 39, 140, 23);
        add(lblNumeroLicencias);

        lblActual = new JLabel("Venta Actual", JLabel.CENTER);
        lblActual.setBounds(15, 77, 441, 23);
        lblActual.setOpaque(true);
        lblActual.setFont(new Font("Dialog", Font.BOLD, 12));
        lblActual.setBackground(new Color(163, 184, 204));
        lblActual.setForeground(Color.black);
        add(lblActual);

        lblHistorico = new JLabel("Reporte Histórico", JLabel.CENTER);
        lblHistorico.setBounds(15, 215, 441, 23);
        lblHistorico.setOpaque(true);
        lblHistorico.setFont(new Font("Dialog", Font.BOLD, 12));
        lblHistorico.setBackground(new Color(163, 184, 204));
        lblHistorico.setForeground(Color.black);
        add(lblHistorico);

        txtS1 = new JTextArea();

```

```

        txtS1.setFont(new Font("monospaced", Font.PLAIN, 12));

        scpScroll11 = new JScrollPane(txtS1);
        scpScroll11.setBounds(15, 100, 441, 100);
        add(scpScroll11);

        txtS2 = new JTextArea();
        txtS2.setFont(new Font("monospaced", Font.PLAIN, 12));

        scpScroll12 = new JScrollPane(txtS2);
        scpScroll12.setBounds(15, 238, 441, 200);
        add(scpScroll12);
    }
    //-----
    // Procesa eventos de tipo(ActionEvent)
    public void actionPerformed( ActionEvent e ){
        if( e.getSource() == btnProcesar )
            procesar();
        if( e.getSource() == btnBorrar )
            borrar();
    }
    // -----
    // Método del botón Borrar
    void borrar(){
        txtNumeroLicencias.setText("");
        txtS1.setText("");
        cboTipo.setSelectedIndex(0);
        txtNumeroLicencias.requestFocus();
    }
    // -----
    // Método del botón Procesar
    void procesar(){
        int tipo, numero;
        double imppag;

        tipo = getTipo();
        numero = getNumero();
        imppag = calcularImportePagar(tipo, numero);
        actualizarDatosHistoricos(tipo, numero, imppag);
        reportarVentaActual(tipo, numero, imppag);
        reportarDatosHistoricos();
    }
    // -----
    // Lee y retorna el tipo de licencia
    int getTipo(){
        return cboTipo.getSelectedIndex();
    }
    // -----
    // Lee y retorna el número de licencias
    int getNumero(){
        return Integer.parseInt(txtNumeroLicencias.getText());
    }
    // -----
    // Calcula y retorna el importe a pagar
    double calcularImportePagar(int tipo, int numero){
        switch(tipo){
            case 0 : return 510*numero;
            case 1 : return 1500*numero;
            case 2 : return 3100*numero;
            default: return 4500*numero;
        }
    }
    // -----
    // Actualiza los datos históricos
    void actualizarDatosHistoricos(int tipo, int numero, double imppag){
        switch(tipo){
            case 0:
                imptot1 += imppag;

```

```

        nlic1 += numero;
        nven1++;
        break;
    case 1:
        imptot2 += imppag;
        nlic2 += numero;
        nven2++;
        break;
    case 2:
        imptot3 += imppag;
        nlic3 += numero;
        nven3++;
        break;
    default:
        imptot4 += imppag;
        nlic4 += numero;
        nven4++;
    }
}
// -----
// Muestra el reporte de la venta actual
void reportarVentaActual(int tipo, int numero, double imppag){
    String strTipo = (String)cboTipo.getItemAt(tipo);
    txtS1.setText("");
    imprimir1("Tipo de licencia.....: " + strTipo);
    imprimir1("Número de licencias.....: " + numero);
    imprimir1("Importe a pagar.....: " + imppag);
}
// -----
// Muestra el reporte de los datos histórico
void reportarDatosHistoricos(){
    txtS2.setText("");
    imprimir2("Importe total recaudado");
    imprimir2("    Por licencias Cobre....: " + imptot1);
    imprimir2("    Por licencias Bronze....: " + imptot2);
    imprimir2("    Por licencias Silver....: " + imptot3);
    imprimir2("    Por licencias Gold.....: " + imptot4);
    imprimir2("");
    imprimir2("Número de licencias vendidas");
    imprimir2("    Por licencias Cobre....: " + nlic1);
    imprimir2("    Por licencias Bronze....: " + nlic2);
    imprimir2("    Por licencias Silver....: " + nlic3);
    imprimir2("    Por licencias Gold.....: " + nlic4);
    imprimir2("");
    imprimir2("Número de ventas efectuadas");
    imprimir2("    Por licencias Cobre....: " + nven1);
    imprimir2("    Por licencias Bronze....: " + nven2);
    imprimir2("    Por licencias Silver....: " + nven3);
    imprimir2("    Por licencias Gold.....: " + nven4);
}
// -----
// Imprime una cadena con un salto de línea en txtS1
void imprimir1(String cad){
    txtS1.append(cad + "\n");
}
// -----
// Imprime una cadena con un salto de línea en txtS2
void imprimir2(String cad){
    txtS2.append(cad + "\n");
}
}

```

Problemas propuestos

Actividad

1. Una tienda vende tres tipos de productos a los precios unitarios dados en la siguiente tabla:

Producto	Precio
A	21.5
B	30.0
C	15.0

Diseñe un programa que permita efectuar ventas y muestre luego de cada venta:

- El importe a pagar.
- El número de ventas efectuadas de cada tipo de producto.
- El importe total recaudado por cada tipo de producto.

Declare como globales a las variables absolutamente necesarias.

2. Diseñe un programa que permita ingresar de una en una las edades de un conjunto de personas y muestre luego de cada ingreso:

- El número de personas mayores de edad.
- El número de personas menores de edad.
- La mayor edad ingresada.
- La menor edad ingresada.

Declare como globales a las variables absolutamente necesarias.

3. Diseñe un programa que permita registrar una donación para uno de tres centros de ayuda social: un comedor de niños, una posta médica o una escuela infantil. El programa mostrará luego de cada donación:

- El número de donantes por cada centro de ayuda social.
- El monto total donado por cada centro de ayuda social.

Centro	Número de donantes	Monto total donado
Comedor	20	4500.0
Posta	10	1200.0
Escuela	80	5750.5

Declare como globales a las variables absolutamente necesarias.

Autoevaluación

1. Diseñe un programa para encuestar a un grupo de personas sobre sus preferencias en bebidas gaseosas entre: Pepsi Cola, Coca Cola y Fanta. La empresa encuestadora desea saber lo siguiente:

- Cuántas personas, entre varones y mujeres, prefieren cada tipo de gaseosa.
- Cuántos varones prefieren cada tipo de gaseosa.
- Cuántas mujeres prefieren cada tipo de gaseosa.

Gaseosa	Varones	Mujeres	Total
Pepsi	46	60	106
Coca	80	50	130
Fanta	100	67	167

Declare como globales a las variables absolutamente necesarias.

2. Un hotel turístico tiene tres tipos de habitaciones, cuyos costos por día se dan en la tabla adjunta. Como oferta, el hotel ofrece un descuento del 15% del importe bruto para más de 5 días de hospedaje y 10% en caso contrario. Diseñe un programa que determine el importe bruto, el importe del descuento y el importe a pagar por parte de un cliente. Además, el programa debe mostrar información actualizada sobre el número de clientes por tipo de habitación y el importe total pagado para los clientes ingresados hasta el momento.

Categoría	Costo por día
A	S/. 90
B	S/. 80
C	S/. 70

El reporte será como el siguiente:

Importe bruto	:	S/. 640
Importe de descuento	:	S/. 96
Importe a pagar	:	S/. 544
Importe total pagado	:	S/. 35867
Clientes habitación A:		12
Clientes habitación B:		8
Clientes habitación C:		15

Declare como globales a las variables absolutamente necesarias.

3. En un instituto, los alumnos se categorizan de acuerdo a su promedio ponderado del ciclo anterior de la siguiente forma:

Promedio	Categoría
17 a 20	A
14 a 16.99	B
11 a 13.99	C
0 a 10.99	D

Diseñe un programa que permita ingresar el promedio ponderado de un alumno por vez y obtener luego de cada ingreso la siguiente información:

- Número de alumnos ingresados.
- Número de alumnos por categoría.
- El mayor promedio ponderado.
- El menor promedio ponderado.

El reporte será como el siguiente:

Número de ingresos	:	120
Alumnos de la categoría A:		40
Alumnos de la categoría B:		20
Alumnos de la categoría C:		25
Alumnos de la categoría D:		35
Mayor promedio ponderado	:	18.5
Menor promedio ponderado	:	11.2

Declare como globales a las variables absolutamente necesarias.

4. Una persona tiene registrado en un papel las compras que efectuó sobre los productos A, B, C y D a lo largo de todo un año. Por cada registro, tiene anotado el tipo del producto adquirido, el precio unitario del producto y la cantidad de unidades adquiridas. Diseñe un programa que permita ingresar, mediante un botón y de uno en uno, los registros con que cuenta la persona y muestre luego de cada ingreso:

- El número de registros de cada tipo de producto.
- La cantidad total de unidades adquiridas de cada tipo de producto.
- El precio unitario promedio de cada tipo de producto.

Declare como globales a las variables absolutamente necesarias.

Producto	Nro. registros	Total unidades	Precio promedio
A	120	300	17.5
B	100	100	18.2
C	80	25	19.5
D	67	67	20.4

Para recordar

- Las variables locales tienen alcance de método; en cambio, las variables globales tienen alcance de programa.
- El tiempo de vida de una variable local es el tiempo que dure la ejecución del método que lo declara; en cambio, el tiempo de vida de una variable global es el tiempo que dura la ejecución del programa.
- Las variables locales no se inicializan automáticamente, cosa que si sucede con las variables globales.

UNIDAD DE
APRENDIZAJE

6

SEMANA

13-16

ESTRUCTURAS DE REPETICIÓN

LOGRO DE LA UNIDAD DE APRENDIZAJE

Al finalizar la unidad, los alumnos, utilizando estructuras de repetición simple y anidada y los demás conceptos aprendidos en el curso, diseñarán algoritmos que resuelvan problemas que involucren instrucciones repetitivas.

CONTENIDO

1. Introducción
2. Estructura de repetición while.
3. Números aleatorios
4. Estructura de repitición do...while.
5. Estructura de repetición for.
6. Problemas resueltos.

ACTIVIDAD

Los alumnos desarrollan algoritmos que involucren el uso de estructuras de repetición.

1. INTRODUCCIÓN

Se denominan **estructuras repetitivas** a aquellas estructuras que permiten repetir instrucciones. A las estructuras repetitivas se conocen también como **estructuras iterativas** o **bucles**, a las instrucciones a repetir se conocen como el **cuerpo del bucle** y al hecho de repetir la secuencia de instrucciones se denomina **iteración**. En el caso del lenguaje Java tenemos tres tipos de estructuras repetitivas: las estructuras **while**, **do...while** y **for**.

2. ESTRUCTURA DE REPETICIÓN while

La estructura **while** repite una acción o un conjunto de acciones mientras sea verdadera una determinada condición, para lo cual primero verifica la condición y luego ejecuta la acción. La acción puede ser una **acción simple** o una **acción compuesta** (bloque de acciones encerradas entre llaves). En las figuras 10.1 se muestra el diagrama de flujo de la estructura **while**.

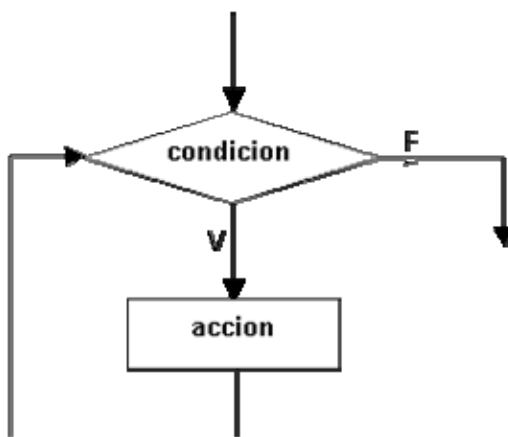


Figura 10.1 Diagrama de flujo de la estructura **while**

La sintaxis de la estructura de repetición **while** es la siguiente:

Para una sólo acción a repetir:

```
while( condicion )
    accion;
```

Para más de una acción a repetir:

```
while( condicion ){
    accion1;
    accion2;
    .
    .
    .
    accion3;
}
```

3. NUMEROS ALEATORIOS ENTEROS

Para generar un número aleatorio entero en el intervalo de **min** a **max** se usa la siguiente expresión:

```
n =(int) ((max-min+1)*Math.random() + min);
```

4. ESTRUCTURA DE REPETICIÓN **do...while**

La estructura **do...while** repite una acción o un conjunto de acciones mientras sea verdadera una determinada condición, para lo cual primero ejecuta la acción y luego verifica la condición. La acción puede ser una **acción simple** o una **acción compuesta** (bloque de acciones encerradas entre llaves). En la figura 10.2 se muestra el diagrama de flujo de la estructura do-while.

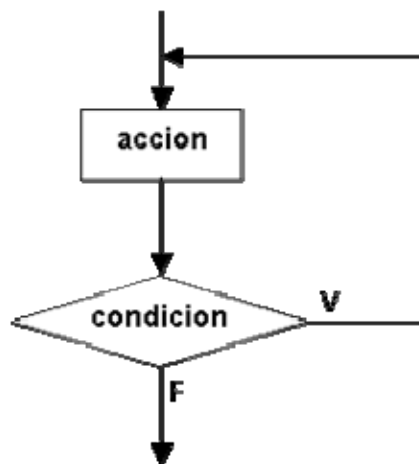


Figura 10.2 Diagrama de flujo de la estructura **do...while**

La sintaxis de la estructura de repetición **do...while** es la siguiente:

Para una sólo acción a repetir:

```
do
    accion;
while( condicion );
```

Para más de una acción a repetir

```
do{
    accion1;
    accion2;
    .
    .
    .
    accion3;
}while( condicion );
```

5. ESTRUCTURA DE REPETICIÓN **for**

La estructura de repetición **for** contempla todos los elementos de la *repetición controlada por contador* (repetición en la que se conoce el número de veces que se repetirá el cuerpo del bucle) al incluir un contador como parte de la propia estructura. En la figura 10.5 se muestra el diagrama de flujo correspondiente:

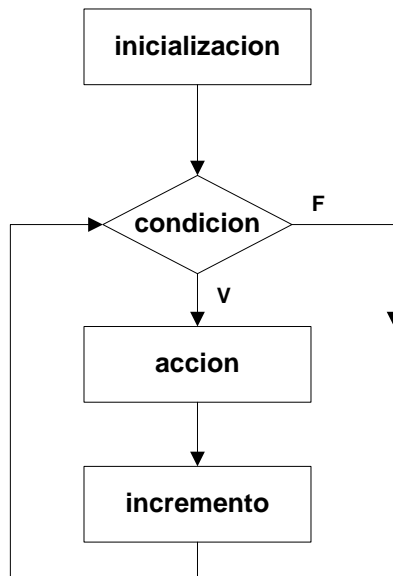


Figura 10.3 Diagrama de flujo de la estructura **for**

La sintaxis de la estructura de repetición **for** es la siguiente:

Para una sólo acción a repetir:

```
for (inicio; condicion; incremento)
    accion;
```

Para más de una acción a repetir:

```
for (inicio; condicion; incremento) {
    accion1;
    accion2;
    .
    .
    .
    accionn;
}
```

Ejemplo 10.1

a. **for** que imprime los números: 0, 1, 2, 3, ..., 48, 49, 50


```
for( int i = 0; i <= 50; i++ )  
    txtS.append( i + "\n" );
```

- b. **for** que imprime los números: 100, 99, 98, ..., 13, 12, 11, 10

```
for( int i = 100; i >= 10; i-- )  
    txtS.append( i + "\n" );
```

- c. **for** que imprime los números: 10, 12, 14, 16, ..., 98, 99, 100

```
for( int i = 10; i <= 100; i += 2 )  
    txtS.append( i + "\n" );
```

- d. **for** que imprime los números: 100, 97, 94, 91, ..., 18, 15, 12, 9

```
for( int i = 100; i >= 9; i -= 3 )  
    txtS.append( i + "\n" );
```

- e. **for** que imprime la siguiente tabla:

10	30
11	29
12	28
13	27
14	26
15	25
16	24
17	23
18	22
19	21
20	20

```
for( int i = 10, j = 30; i <= 20; i++, j-- )  
    txtS.append( i + "\t" + j + "\n" );
```

Nota

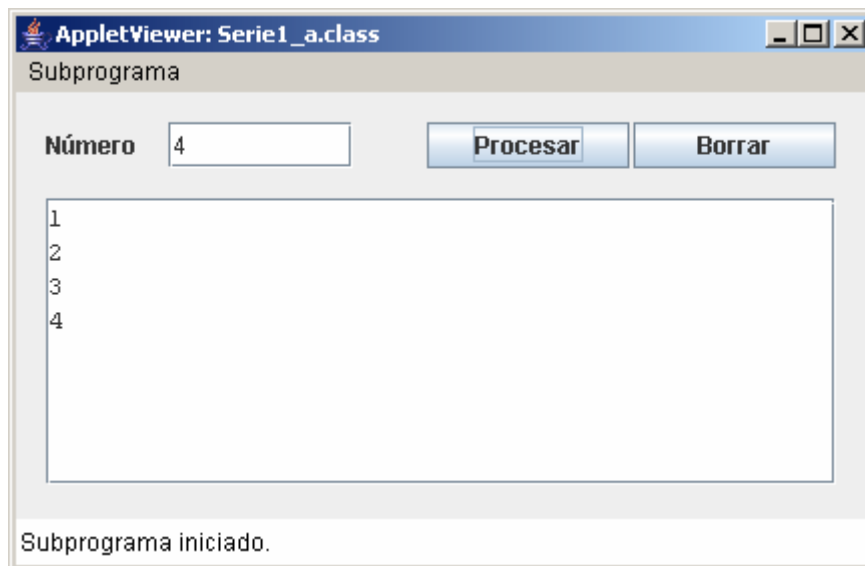
El contador del **for** puede ser declarado dentro del mismo **for**, en cuyo caso se considera como variable local al **for**, no siendo accesible fuera de él. De esta manera, en un mismo método dos o más **for**, no anidados, pueden declarar contadores con el mismo nombre. Cada contador existe dentro del **for** en el que fue declarado.

6. PROBLEMAS RESUELTOS

Problema 10.1

Diseña un programa que imprima la siguiente serie:

1, 2, 3, ..., n



Solución 1 : usando while

```

import java.awt.event.*;
import java.awt.*;
import javax.swing.*;

public class Serie1_a extends JApplet implements ActionListener{
    // Declaración de variables globales
    JButton btnProcesar, btnBorrar;
    JTextArea txtS;
    JLabel lblNumero;
    JTextField txtNumero;
    JScrollPane scpScroll;

    // -----
    // Crea la interfaz gráfica de usuario GUI
    public void init() {
        getContentPane().setLayout(null);

        txtNumero = new JTextField();
        txtNumero.setBounds(76, 15, 92, 23);
        getContentPane().add(txtNumero);

        btnBorrar = new JButton("Borrar");
        btnBorrar.setBounds(308, 15, 101, 23);
        btnBorrar.addActionListener(this);
        getContentPane().add(btnBorrar);

        btnProcesar = new JButton("Procesar");
        btnProcesar.setBounds(205, 15, 101, 23);
        btnProcesar.addActionListener(this);
        getContentPane().add(btnProcesar);

        lblNumero = new JLabel("Número");
        lblNumero.setBounds(15, 15, 56, 23);
        getContentPane().add(lblNumero);

        txtS = new JTextArea();
        txtS.setFont(new Font("monospaced", 0, 12));

        scpScroll = new JScrollPane(txtS);
        scpScroll.setBounds(15, 53, 394, 143);
        getContentPane().add(scpScroll);
    }
    // -----
    // Procesa eventos de tipo ActionEvent
    public void actionPerformed( ActionEvent e ){
        if( e.getSource() == btnProcesar )
            procesar();
        if( e.getSource() == btnBorrar )
            borrar();
    }
    // -----
    // Método del botón Borrar
    void borrar(){
        txtNumero.setText("");
        txtS.setText("");
        txtNumero.requestFocus();
    }
    // -----
    // Método del botón Procesar
    void procesar(){

        // Declaración e inicialización de variables locales
        int n, termino = 1;

        // Lee el número n
        n = Integer.parseInt(txtNumero.getText());

```

```

// Borra la pantalla
txtS.setText("");

// Bucle while
while( termino <= n ){

    // Imprime el término actual
    txtS.append(termino + "\n");

    // Genera el siguiente término
    termino++;

}
}

```

Solución 2 : usando do...while

```

// Método del botón Procesar
void procesar(){

    // Declaración e inicialización de variables locales
    int n, termino = 1;

    // Lee el número n
    n = Integer.parseInt(txtNumero.getText());

    // Borrar la pantalla
    txtS.setText("");

    // Bucle do...while
    do{
        // Imprime el término actual
        txtS.append(termino + "\n");

        // Genera el siguiente término
        termino++;
    }while( termino <= n );
}

```

Solución 3: usando for

```

// Método del botón Procesar
void procesar(){

    // Declaración de variables locales
    int n, termino = 1;

    // Lee el número n
    n = Integer.parseInt(txtNumero.getText());

    // Borra la pantalla
    txtS.setText("");

    // Bucle for
    for( termino = 1; termino <= n; termino ++ )

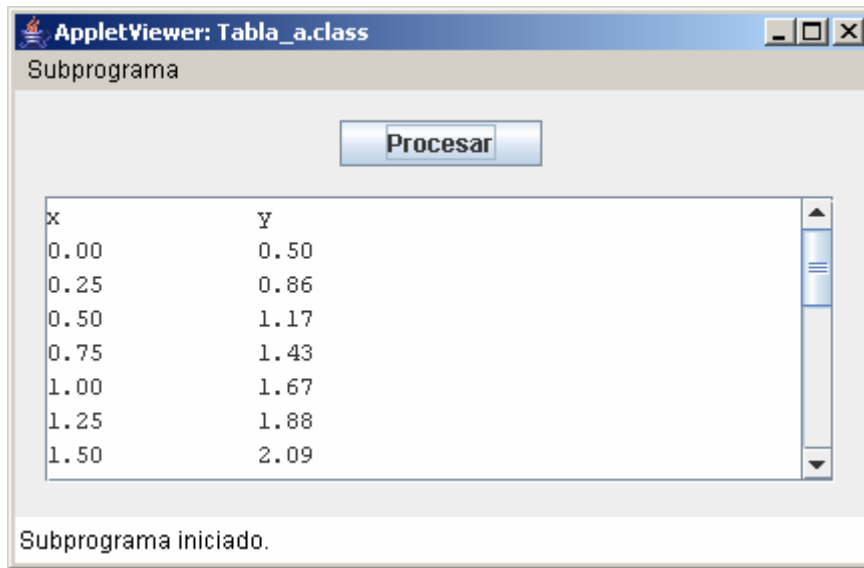
        // Imprime el término actual
        txtS.append(termino + "\n");
}

```

Problema 10.2

Diseña un programa que imprima una tabla de valores de x e y, para valores de x en el intervalo de 0 a 10 cada 0.25, siendo:

$$y = \frac{x^3 + 3x + 1}{x^2 + 2}$$



Solución 1: solución con while

```
import java.awt.event.*;
import java.awt.*;
import javax.swing.*;
import java.util.*;

public class Tabla_a extends JApplet implements ActionListener{
    // Declaración de variables globales
    JButton btnProcesar;
    JTextArea txtS;
    JScrollPane scpScroll;

    // -----
    // Crea la interfaz gráfica de usuario GUI
    public void init() {
        getContentPane().setLayout(null);

        btnProcesar = new JButton("Procesar");
        btnProcesar.setBounds(162, 15, 101, 23);
        btnProcesar.addActionListener(this);
        getContentPane().add(btnProcesar);

        txtS = new JTextArea();
        txtS.setFont(new java.awt.Font("monospaced", 0, 12));

        scpScroll = new JScrollPane(txtS);
        scpScroll.setBounds(15, 53, 394, 143);
        getContentPane().add(scpScroll);
    }
    // -----
    // Procesa eventos de tipo(ActionEvent)
    public void actionPerformed( ActionEvent e ){
        if( e.getSource() == btnProcesar )
            procesar();
    }
    // -----
    // Método del botón Procesar
    void procesar(){
```

```

// Declaración e inicialización de variables
double x = 0, y;

// Imprime la cabecera
txtS.append(String.format(Locale.US, "%-15.2s%-15.2s\n", "x", "y"));

// Imprime la tabla
while( x <= 5 ){

    // Calcula el valor de y para el valor actual de x
    y = (x*x*x + 3*x + 1)/(x*x + 2);

    // Imprime la pareja de valores x e y
    txtS.append(String.format(Locale.US, "%-15.2f%-15.2f\n", x, y));

    // Incrementa x en 0.25
    x += 0.25;
}
}

```

Solución 2: solución do...while

```

// Método del botón Procesar
void procesar(){

    // Declaración e inicialización de variables
    double x = 0, y;

    // Imprime la cabecera
    txtS.append(String.format(Locale.US, "%-15.2s%-15.2s\n", "x", "y"));

    // Imprime la tabla
    do{

        // Calcula el valor de y para el valor actual de x
        y = (x*x*x + 3*x + 1)/(x*x + 2);

        // Imprime la pareja de valores x e y
        txtS.append(String.format(Locale.US, "%-15.2f%-15.2f\n", x, y));

        // Incrementa x en 0.25
        x += 0.25;
    }while( x <= 5 );
}

```

Solución 3: solución con for

```

// Método del botón Procesar
void procesar(){

    // Declaración e inicialización de variables
    double y;

    // Imprime la cabecera
    txtS.append(String.format(Locale.US, "%-15.2s%-15.2s\n", "x", "y"));

    // Imprime la tabla
    for( double x = 0; x <= 5; x += 0.25 ){

        // Calcula el valor de y para el valor actual de x
        y = (x*x*x + 3*x + 1)/(x*x + 2);

        // Imprime la pareja de valores x e y
        txtS.append(String.format(Locale.US, "%-15.2f%-15.2f\n", x, y));
    }
}

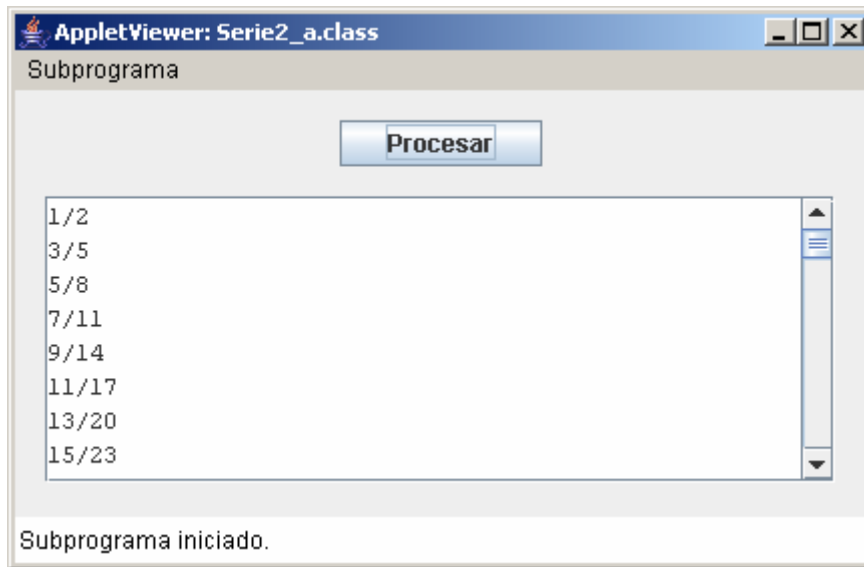
```

}

Problema 10.3

Diseñe un programa que imprima la siguiente serie y halle su suma:

$$1/2 + 3/5 + 5/8 + 7/11 + \dots \text{ (100 términos)}$$

**Solución 1: solución con while**

```
import java.awt.event.*;
import java.awt.*;
import javax.swing.*;

public class Serie2_a extends JApplet implements ActionListener{
    // Declaración de variables globales
    JButton btnProcesar;
    JTextArea txtS;
    JScrollPane scpScroll;

    // -----
    // Crea la interfaz gráfica de usuario GUI
    public void init() {
        getContentPane().setLayout(null);

        btnProcesar = new JButton("Procesar");
        btnProcesar.setBounds(162, 15, 101, 23);
        btnProcesar.addActionListener(this);
        getContentPane().add(btnProcesar);

        txtS = new JTextArea();
        txtS.setFont(new java.awt.Font("monospaced", 0, 12));

        scpScroll = new JScrollPane(txtS);
        scpScroll.setBounds(15, 53, 394, 143);
        getContentPane().add(scpScroll);
    }
    // -----
    // Procesa eventos de tipo ActionEvent
    public void actionPerformed( ActionEvent e ){
        if( e.getSource() == btnProcesar )
            procesar();
    }
}
```

```
// -----
// Método del botón Procesar
void procesar(){

    // Declaración de variables locales
    int numerador = 1, denominador = 2, contador = 0;
    double suma = 0;

    // Genera e imprime la serie
    while( contador < 100 ){

        // Imprime el término actual de la serie
        txtS.append(numerador + "/" + denominador + "\n");

        // Suma el término actual
        suma += numerador*1.0/denominador;

        // Genera el numerador y el denominador de un nuevo término
        numerador += 2;
        denominador += 3;

        // Incrementa el contador de términos
        contador++;

    }

    // Imprime la suma
    txtS.append( "\nSuma : " + suma );

}
}
```

Solución 2: solución con do...while

```
// Método del botón Procesar
void procesar(){

    // Declaración de variables locales
    int numerador = 1, denominador = 2, contador = 0;
    double suma = 0;

    // Genera e imprime la serie
    do{

        // Imprime el término actual de la serie
        txtS.append(numerador + "/" + denominador + "\n");

        // Suma el término actual
        suma += numerador*1.0/denominador;

        // Genera el numerador y el denominador de un nuevo término
        numerador += 2;
        denominador += 3;

        // Incrementa el contador de términos
        contador++;

    }while( contador < 100 );

    // Imprime la suma
    txtS.append( "\nSuma : " + suma );

}
}
```

Solución 3: solución con for

```
// Método del botón Procesar
void procesar(){
```



```
// Declaración de variables locales
int numerador = 1, denominador = 2;
double suma = 0;

// Genera e imprime la serie
for( int contador = 0; contador < 100; contador++ ){

    // Imprime el término actual de la serie
    txtS.append(numerador + "/" + denominador + "\n");

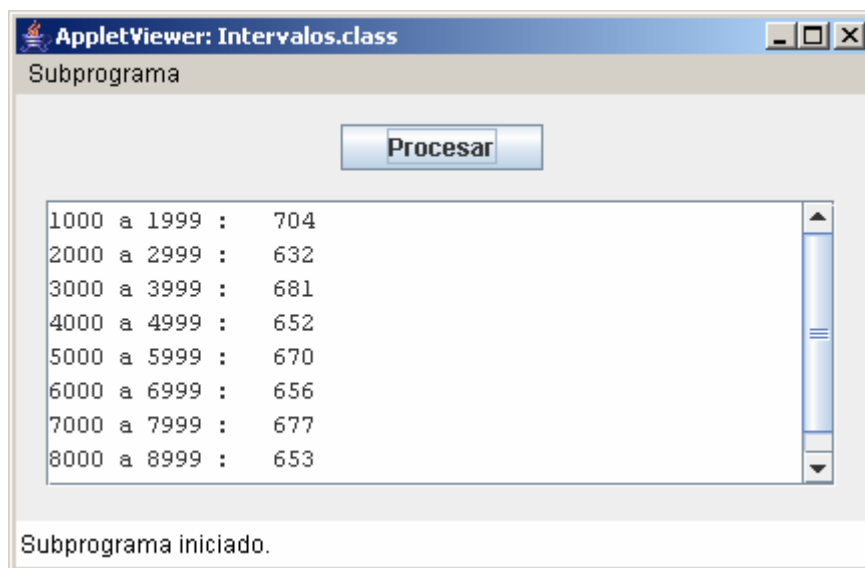
    // Suma el término actual
    suma += numerador*1.0/denominador;

    // Genera el numerador y el denominador de un nuevo término
    numerador += 2;
    denominador += 3;
}

// Imprime la suma
txtS.append( "\nSuma : " + suma );
}
```

Problema 10.4

Diseñe un programa que genere 6000 números aleatorios en el intervalo de 1000 a 9999 y determine cuántos números están en cada uno de los siguientes intervalos: 1000 a 1999, 2000 a 2999, 3000 a 3999, ..., 8000 a 8999, 9000 a 9999.



Solución

```
import java.awt.event.*;
import java.awt.*;
import javax.swing.*;

public class Intervalos extends JApplet implements ActionListener{
    // Declaración de variables globales
    JButton btnProcesar;
    JTextArea txtS;
    JScrollPane scpScroll;

    // -----
    // Crea la interfaz gráfica de usuario GUI
}
```

```

public void init() {
    getContentPane().setLayout(null);

    btnProcesar = new JButton("Procesar");
    btnProcesar.setBounds(162, 15, 101, 23);
    btnProcesar.addActionListener(this);
    getContentPane().add(btnProcesar);

    txtS = new JTextArea();
    txtS.setFont(new java.awt.Font("monospaced", 0, 12));

    // scpScroll
    scpScroll = new JScrollPane(txtS);
    scpScroll.setBounds(15, 53, 394, 143);
    getContentPane().add(scpScroll);
}
// -----
// Procesa eventos de tipo ActionEvent
public void actionPerformed( ActionEvent e ){
    if( e.getSource() == btnProcesar )
        procesar();
}
// -----
// Imprime una línea de texto con un salto de línea
void imprimir(String cad){
    txtS.append(cad + "\n");
}
// -----
// Método del botón Procesar
void procesar(){

    // Declaraciones locales
    int numero, c1=0, c2=0, c3=0, c4=0, c5=0, c6=0, c7=0, c8=0, c9=0;

    // Genera los 600 números aleatorios
    for( int i = 0; i < 6000; i++ ){

        // Genera un número de 1000 a 9999
        numero = (int)((9999-1000+1)*Math.random()+1000);

        // Cuenta el número según el intervalo
        if( numero < 2000 )
            c1++;
        else if( numero < 3000 )
            c2++;
        else if( numero < 4000 )
            c3++;
        else if( numero < 5000 )
            c4++;
        else if( numero < 6000 )
            c5++;
        else if( numero < 7000 )
            c6++;
        else if( numero < 8000 )
            c7++;
        else if( numero < 9000 )
            c8++;
        else
            c9++;
    }

    // Imprime los contadores
    txtS.setText("");
    imprimir("1000 a 1999 : \t" + c1);
    imprimir("2000 a 2999 : \t" + c2);
    imprimir("3000 a 3999 : \t" + c3);
    imprimir("4000 a 4999 : \t" + c4);
    imprimir("5000 a 5999 : \t" + c5);
}

```

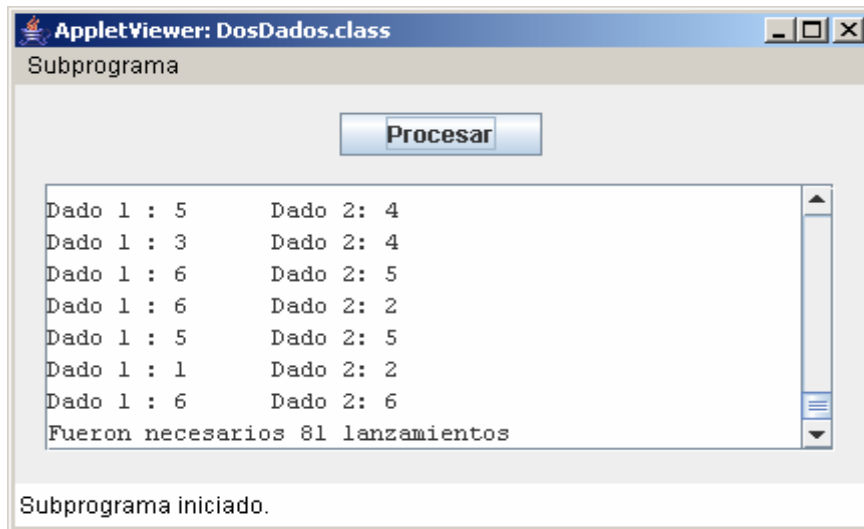
```

        imprimir("6000 a 6999 : \t" + c6);
        imprimir("7000 a 7999 : \t" + c7);
        imprimir("8000 a 8999 : \t" + c8);
        imprimir("9000 a 9999 : \t" + c9);
    }
}

```

Problema 10.5

Diseñe un programa que simule varios lanzamientos simultáneos de dos dados hasta obtener 6 en ambos dados e informe cuántos lanzamientos simultáneos fueron necesarios efectuar.



Solución

```

import java.awt.event.*;
import java.awt.*;
import javax.swing.*;

public class DosDados extends JApplet implements ActionListener{
    // Declaración de variables globales
    JButton btnProcesar;
    JTextArea txtS;
    JScrollPane scpScroll;

    // -----
    // Crea la interfaz gráfica de usuario GUI
    public void init() {
        getContentPane().setLayout(null);

        btnProcesar = new JButton("Procesar");
        btnProcesar.setBounds(162, 15, 101, 23);
        btnProcesar.addActionListener(this);
        getContentPane().add(btnProcesar);

        txtS = new JTextArea();
        txtS.setFont(new java.awt.Font("monospaced", 0, 12));

        scpScroll = new JScrollPane(txtS);
        scpScroll.setBounds(15, 53, 394, 143);
        getContentPane().add(scpScroll);
    }
    // -----
    // Procesa eventos de tipo(ActionEvent)
    public void actionPerformed( ActionEvent e ){

```

```

        if( e.getSource() == btnProcesar )
            procesar();
    }
    // -----
    // Imprime una línea de texto con un salto de línea
    void imprimir(String cad){
        txtS.append(cad + "\n");
    }
    // -----
    // Método del botón procesar
    void procesar(){
        // Declaración de variables locales
        int dado1, dado2, conta=0;

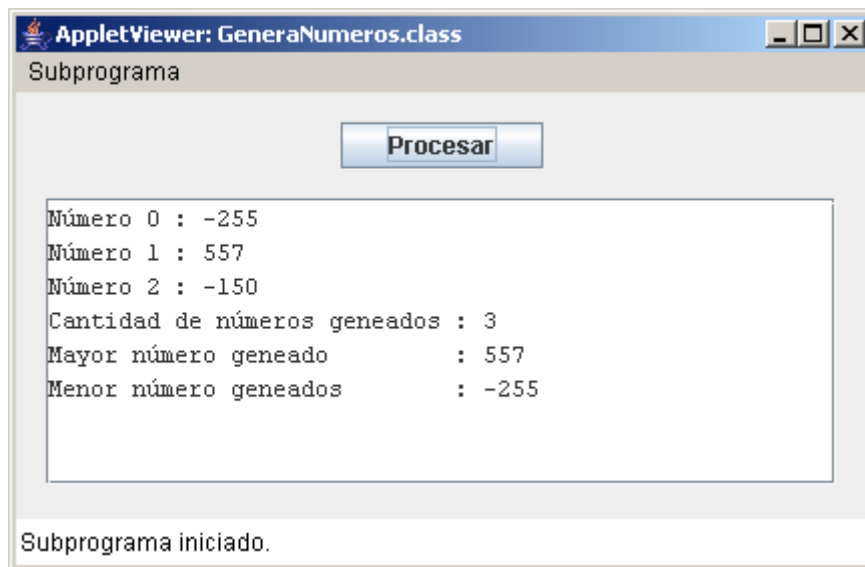
        // Simula el lanzamiento de los dados
        txtS.setText("");
        do{
            dado1 = (int)(6*Math.random()+1);
            dado2 = (int)(6*Math.random()+1);
            imprimir("Dado 1 : " + dado1 + "\tDado 2: " + dado2);
            conta++;
        }while(dado1 != 6 || dado2 != 6);

        // Imprime la cantidad de lanzamientos efectuados
        imprimir("Fueron necesarios " + conta + " lanzamientos");
    }
}

```

Problema 10.6

Diseñe un programa que genere números aleatorios en el intervalo de -1000 a 1000 hasta obtener un número par menor que 500 pero mayor que -500. ¿Cuántos números fueron necesarios generar y cuáles fueron el mayor y el menor número generados?.



Solución

```

import java.awt.event.*;
import java.awt.*;
import javax.swing.*;

public class GeneraNumeros extends JApplet implements ActionListener{
    // Declaración de variables globales
    JButton btnProcesar;
    JTextArea txtS;
}

```

```

JScrollPane scpScroll;

// -----
// Crea la interfaz gráfica de usuario GUI
public void init() {
    getContentPane().setLayout(null);

    btnProcesar = new JButton("Procesar");
    btnProcesar.setBounds(162, 15, 101, 23);
    btnProcesar.addActionListener(this);
    getContentPane().add(btnProcesar);

    txtS = new JTextArea();
    txtS.setFont(new java.awt.Font("monospaced", 0, 12));

    scpScroll = new JScrollPane(txtS);
    scpScroll.setBounds(15, 53, 394, 143);
    getContentPane().add(scpScroll);
}
// -----
// Procesa eventos de tipo ActionEvent
public void actionPerformed( ActionEvent e ){
    if( e.getSource() == btnProcesar )
        procesar();
}
// -----
// Imprime una línea de texto con un salto de línea
void imprimir(String cad){
    txtS.append(cad + "\n");
}
// -----
// Método para el botón Procesar
void procesar(){

    // Declaración de variables locales
    int numero, mayor = -1000, menor = 1000, conta = 0;

    // Generación de números
    txtS.setText("");
    do{

        // Genera un número aleatorio en el intervalo de -1000 y 1000
        numero = (int) (2001*Math.random()-1000);

        // Muestra y cuenta el número generado
        imprimir("Número " + conta + " : " + numero);
        conta++;

        // Actualiza el mayor y el menor
        if( numero > mayor )
            mayor = numero;
        if( numero < menor )
            menor = numero;
    }while( numero%2 != 0 || numero <= -500 || numero >= 500 );

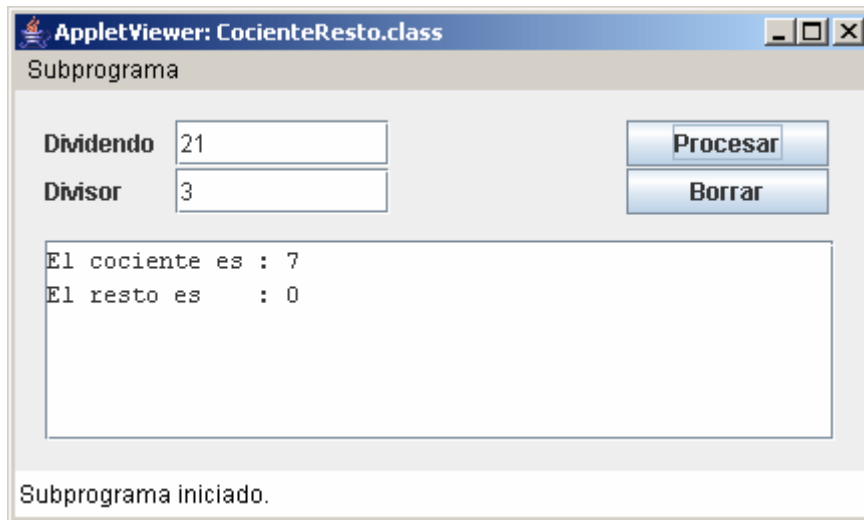
    // Imprime resultados
    imprimir("Cantidad de números generados : " + conta);
    imprimir("Mayor número generado : " + mayor);
    imprimir("Menor número generado : " + menor);
}
}

```

Problema 10.7

Un método para obtener el cociente y el resto de dos números enteros es por restas sucesivas. El método consiste en restar sucesivamente el divisor del dividendo hasta obtener un resultado menor que el divisor, que será el resto de la división; el número

de restas efectuadas será el cociente. Diseñe un algoritmo que determine el cociente y el resto de dos números enteros.



Solución

```
import java.awt.event.*;
import java.awt.*;
import javax.swing.*;

public class CocienteResto extends JApplet implements ActionListener{
    // Declaración de variables globales
    JButton btnProcesar;
    JButton btnBorrar;
    JTextArea txtS;
    JLabel lblDivisor;
    JLabel lblDividendo;
    JTextField txtDivisor;
    JTextField txtDividendo;
    JScrollPane scpScroll;

    // -----
    // Crea la interfaz gráfica de usuario GUI
    public void init() {
        getContentPane().setLayout(null);

        lblDivisor = new JLabel("Divisor");
        lblDivisor.setBounds(15, 39, 58, 23);
        getContentPane().add(lblDivisor);

        lblDividendo = new JLabel("Dividendo");
        lblDividendo.setBounds(15, 15, 58, 23);
        getContentPane().add(lblDividendo);

        txtDividendo = new JTextField();
        txtDividendo.setBounds(80, 15, 107, 23);
        getContentPane().add(txtDividendo);

        txtDivisor = new JTextField();
        txtDivisor.setBounds(80, 39, 107, 23);
        getContentPane().add(txtDivisor);

        btnProcesar = new JButton("Procesar");
        btnProcesar.setBounds(305, 15, 101, 23);
        btnProcesar.addActionListener(this);
        getContentPane().add(btnProcesar);

        btnBorrar = new JButton("Borrar");
```

```

        btnBorrar.setBounds(305,39,101,23);
        btnBorrar.addActionListener(this);
        getContentPane().add(btnBorrar);

        txtS = new JTextArea();
        txtS.setFont(new Font("monospaced", 0, 12));

        scpScroll = new JScrollPane(txtS);
        scpScroll.setBounds(15, 75, 394, 100);
        getContentPane().add(scpScroll);
    }
    // -----
    // Procesa eventos de tipo ActionEvent
    public void actionPerformed( ActionEvent e ){
        if( e.getSource() == btnProcesar )
            procesar();
        if( e.getSource() == btnBorrar )
            borrar();
    }
    // -----
    // Método para el botón Borrar
    void borrar(){
        txtDivisor.setText("");
        txtDividendo.setText("");
        txtDivisor.requestFocus();
    }
    // -----
    // Imprime una línea de texto con un salto de línea
    void imprimir(String cad){
        txtS.append(cad + "\n");
    }
    // -----
    // Método para el botón Procesar
    void procesar(){

        // Declaración de variables locales
        int dividendo, divisor, cociente, resto;

        // Ingreso del Divisor y del Dividendo
        divisor = Integer.parseInt(txtDivisor.getText());
        dividendo = Integer.parseInt(txtDividendo.getText());

        // Inicializa el resto y el cociente
        resto = dividendo;
        cociente = 0;

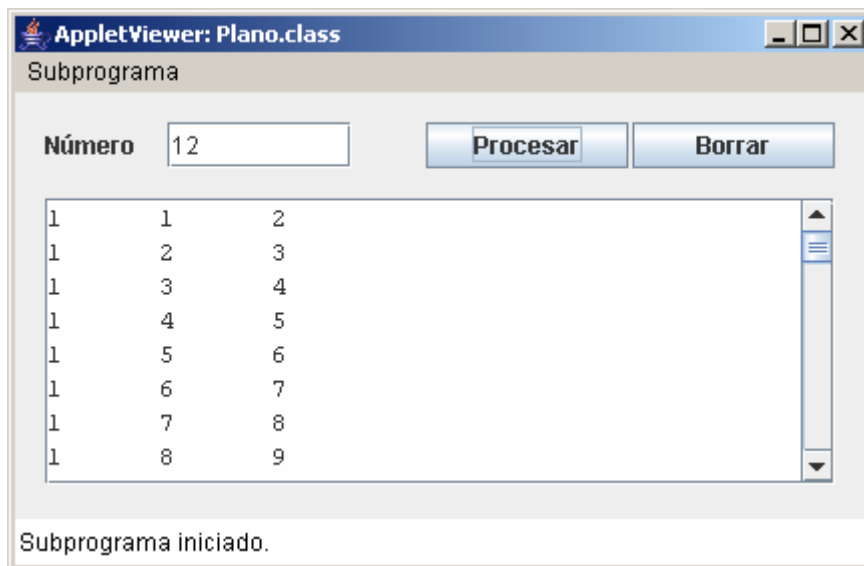
        // Determina el resto y el cociente
        while( resto >= divisor ){
            resto -= divisor;
            cociente++;
        }

        // Imprime el resto y el cociente
        txtS.setText("");
        imprimir("El cociente es : " + cociente);
        imprimir("El resto es      : " + resto);
    }
}

```

Problema 10.8

Diseñe un programa que lea un número entero positivo n mayor que 10 e imprima todos los puntos del plano cartesiano que cumplan con la condición: $x + y < n$.



Solución

```
import java.awt.event.*;
import java.awt.*;
import javax.swing.*;
```

```
public class Plano extends JApplet implements ActionListener{
    // Declaración de variables globales
    JButton btnProcesar, btnBorrar;
    JTextArea txtS;
    JLabel lblNumero;
    JTextField txtNumero;
    JScrollPane scpScroll;

    // -----
    // Crea la interfaz gráfica de usuario GUI
    public void init() {
        getContentPane().setLayout(null);

        txtNumero = new JTextField();
        txtNumero.setBounds(76, 15, 92, 23);
        getContentPane().add(txtNumero);

        btnBorrar = new JButton("Borrar");
        btnBorrar.setBounds(308, 15, 101, 23);
        btnBorrar.addActionListener(this);
        getContentPane().add(btnBorrar);

        btnProcesar = new JButton("Procesar");
        btnProcesar.setBounds(205, 15, 101, 23);
        btnProcesar.addActionListener(this);
        getContentPane().add(btnProcesar);

        lblNumero = new JLabel("Número");
        lblNumero.setBounds(15, 15, 56, 23);
        getContentPane().add(lblNumero);

        txtS = new JTextArea();
        txtS.setFont(new Font("monospaced", 0, 12));

        scpScroll = new JScrollPane(txtS);
        scpScroll.setBounds(15, 53, 394, 143);
        getContentPane().add(scpScroll);
    }
    // -----
    // Procesa eventos de tipo(ActionEvent)
```



```

public void actionPerformed( ActionEvent e ){
    if( e.getSource() == btnProcesar )
        procesar();
    if( e.getSource() == btnBorrar )
        borrar();
}
// -----
// Método del botón borrar
void borrar(){
    txtNumero.setText("");
    txtS.setText("");
    txtNumero.requestFocus();
}
// -----
// Método del botón Procesar
void procesar(){

    // Declaración de variables locales
    int n;

    // Lee el número n
    n = Integer.parseInt(txtNumero.getText());

    // Si n es correcto
    if( n > 10 ){

        // Imprime los puntos del plano que cumplen: x + y < n
        for( int x = 1; x < n; x++ )
            for( int y = 1; x + y < n; y++ )
                txtS.append(x + "\t" + y + "\t" + (x+y) + "\n");
    }
    else
        txtS.setText("El número tiene que ser mayor que 10");
}
}

```

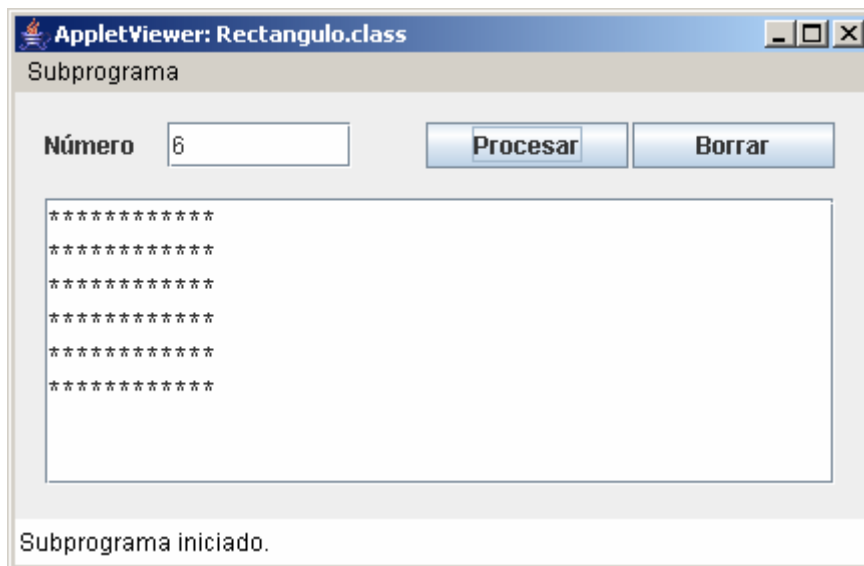
Problema 10.9

Imprime un rectángulo de altura n y ancho $2n$ relleno de asteriscos. Así, para n igual a 5, el programa deberá imprimir un rectángulo de 10 asteriscos de ancho por 5 asteriscos de alto como el que se muestra a continuación:

```

*****
*****
*****
*****
*****

```



Solución

```
import java.awt.event.*;
import java.awt.*;
import javax.swing.*;

public class Rectangulo extends JApplet implements ActionListener{
    // Declaración de variables globales
    JButton btnProcesar, btnBorrar;
    JTextArea txtS;
    JLabel lblNumero;
    JTextField txtNumero;
    JScrollPane scpScroll;

    // -----
    // Crea la interfaz gráfica de usuario GUI
    public void init() {
        getContentPane().setLayout(null);

        txtNumero = new JTextField();
        txtNumero.setBounds(76, 15, 92, 23);
        getContentPane().add(txtNumero);

        btnBorrar = new JButton("Borrar");
        btnBorrar.setBounds(308, 15, 101, 23);
        btnBorrar.addActionListener(this);
        getContentPane().add(btnBorrar);

        btnProcesar = new JButton("Procesar");
        btnProcesar.setBounds(205, 15, 101, 23);
        btnProcesar.addActionListener(this);
        getContentPane().add(btnProcesar);

        lblNumero = new JLabel("Número");
        lblNumero.setBounds(15, 15, 56, 23);
        getContentPane().add(lblNumero);

        txtS = new JTextArea();
        txtS.setFont(new Font("monospaced", 0, 12));

        scpScroll = new JScrollPane(txtS);
        scpScroll.setBounds(15, 53, 394, 143);
        getContentPane().add(scpScroll);
    }
    // -----
    // Procesa eventos de tipo(ActionEvent)
```

```

public void actionPerformed( ActionEvent e ){
    if( e.getSource() == btnProcesar )
        procesar();
    if( e.getSource() == btnBorrar )
        borrar();
}
// -----
// Método del botón Borrar
void borrar(){
    txtNumero.setText("");
    txtS.setText("");
    txtNumero.requestFocus();
}
// -----
// Método del botón Procesar
void procesar(){

    // Declaración de variables locales
    int n;

    // Lee el número n
    n = Integer.parseInt(txtNumero.getText());

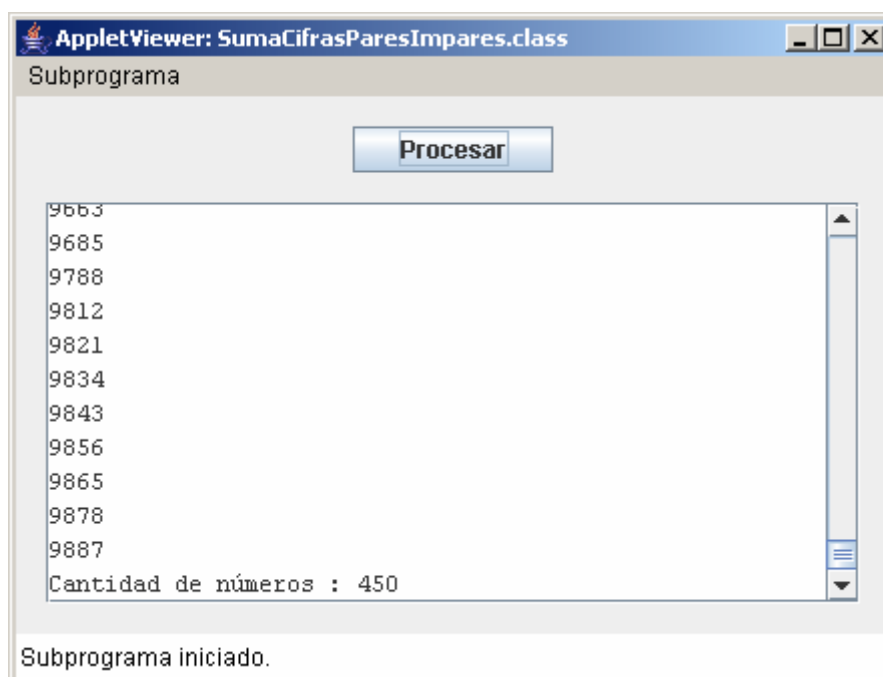
    // Borra el JTextArea por si no fue borrado
    txtS.setText("");

    // Imprime el rectángulo
    for( int i = 0; i < n; i++ ){
        for( int j = 0; j < 2*n; j++ )
            txtS.append("*");
        txtS.append("\n");
    }
}
}

```

Problema 10.10

Diseñe un programa que muestre por pantalla todos los números de 4 cifras que cumplen con la condición de que la suma de las cifras pares es igual a la suma de las cifras impares. Muestre también la cantidad de números encontrados.



Solución

```
import java.awt.event.*;
import java.awt.*;
import javax.swing.*;

public class SumaCifrasParesImpares extends JApplet implements ActionListener{
    // Declaración de variables miembro
    JButton btnProcesar;
    JTextArea txtS;
    JScrollPane scpScroll;

    // -----
    // Crea la interfaz gráfica de usuario
    public void init() {
        getContentPane().setLayout(null);

        btnProcesar = new JButton("Procesar");
        btnProcesar.setBounds(168, 15, 100, 23);
        btnProcesar.addActionListener(this);
        getContentPane().add(btnProcesar);

        txtS = new JTextArea();
        txtS.setFont(new Font("monospaced", 0, 12));

        scpScroll = new JScrollPane(txtS);
        scpScroll.setBounds(15, 53, 406, 200);
        getContentPane().add(scpScroll);
    }
    // -----
    // Procesa eventos de tipo(ActionEvent)
    public void actionPerformed( ActionEvent e ){
        if( e.getSource() == btnProcesar )
            procesar();
    }
    // -----
    // Método del botón Procesar
    void procesar(){

        // Declaración de variables
        int sumaCifPares, sumaCifImpares, cifra, copiaNumero, cantidad=0;

        // Muestra los números solicitados
        txtS.setText("");
        for( int numero=1000; numero<=9999; numero++ ){
            copiaNumero=numero;
            sumaCifImpares=0;
            sumaCifPares=0;
            do{
                cifra = copiaNumero%10;
                if(cifra%2==0)
                    sumaCifPares+=cifra;
                else
                    sumaCifImpares+=cifra;
                copiaNumero /= 10;
            }while(copiaNumero!=0);

            if( sumaCifPares == sumaCifImpares ){
                txtS.append(numero + "\n");
                cantidad++;
            }
        }
        txtS.append("Cantidad de números : " + cantidad);
    }
}
```

Problema 10.11

Diseñe un algoritmo que determine la cantidad de divisores pares de un número natural.

Solución

```
import java.awt.event.*;
import java.awt.*;
import javax.swing.*;

public class CantidadDivisores extends JApplet implements ActionListener{
    // Declaración de variables globales
    JButton btnProcesar, btnBorrar;
    JTextArea txtS;
    JLabel lblNumero;
    JTextField txtNumero;
    JScrollPane scpScroll;

    // -----
    // Crea la interfaz gráfica de usuario
    public void init() {
        getContentPane().setLayout(null);

        lblNumero = new JLabel("Numero");
        lblNumero.setBounds(15, 15, 90, 23);
        getContentPane().add(lblNumero);

        txtNumero = new JTextField();
        txtNumero.setBounds(105, 15, 100, 23);
        getContentPane().add(txtNumero);

        btnProcesar = new JButton("Procesar");
        btnProcesar.setBounds(365, 15, 100, 23);
        btnProcesar.addActionListener(this);
        getContentPane().add(btnProcesar);

        btnBorrar = new JButton("Borrar");
        btnBorrar.setBounds(265, 15, 100, 23);
        btnBorrar.addActionListener(this);
        getContentPane().add(btnBorrar);

        txtS = new JTextArea();
```

```

        txtS.setFont(new Font("monospaced", 0, 12));

        scpScroll = new JScrollPane(txtS);
        scpScroll.setBounds(15, 53, 450, 150);
        getContentPane().add(scpScroll);
    }
    // -----
    // Procesa eventos de tipo(ActionEvent)
    public void actionPerformed( ActionEvent e ){
        if( e.getSource() == btnProcesar )
            procesar();
        if( e.getSource() == btnBorrar )
            borrar();
    }
    // -----
    // Método del botón Borrar
    void borrar(){
        txtNumero.setText("");
        txtS.setText("");
        txtNumero.requestFocus();
    }
    // -----
    // Imprime una línea de texto con un salto de línea
    void imprimir(String cad){
        txtS.append(cad + "\n");
    }
    // -----
    // Método del botón Procesar
    void procesar(){

        // Declaración e inicialización de variables locales
        int contadiv = 0, numero;

        // Ingresar el número
        numero = Integer.parseInt(txtNumero.getText());

        // Determina la cantidad de divisores pares
        for( int divisor = 2; divisor <= numero; divisor += 2 )
            if( numero%divisor == 0 ){
                contadiv++;
                imprimir(divisor + " ");
            }

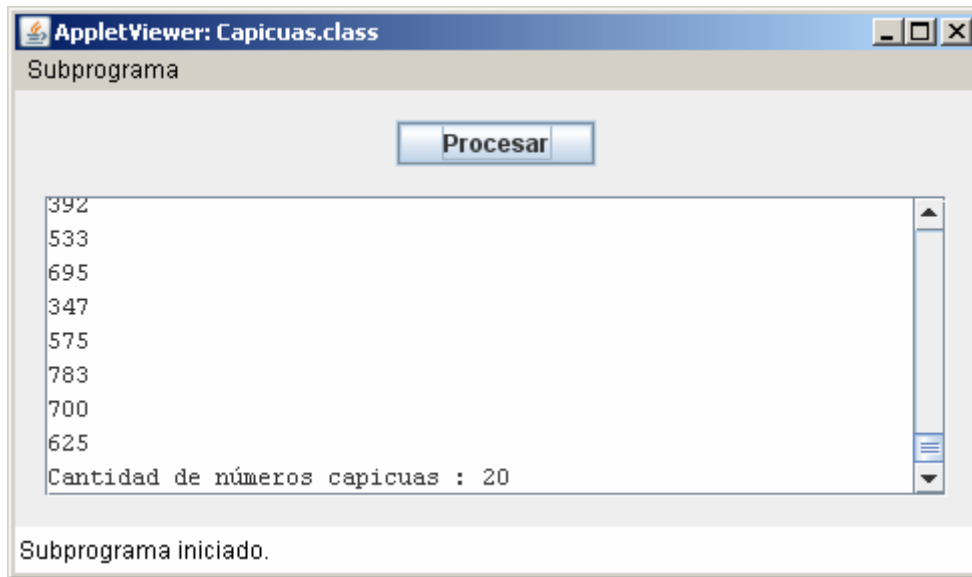
        // Imprime la cantidad de divisores pares
        imprimir("Cantidad de divisores pares : " + contadiv);
    }
}

```

Problema 10.12

Diseña un algoritmo que genere 200 números aleatorios en el intervalo de 100 a 999 y determine la cantidad de números capicúas generados. Un número es capicúa si lee igual de derecha a izquierda que de izquierda a derecha. Así, por ejemplo, 343 es capicúa; pero, 367 no lo es.

Solución



```
import java.awt.event.*;
import java.awt.*;
import javax.swing.*;
```

```
public class Capicuas extends JApplet implements ActionListener{
    // Declaración de variables globales
    JButton btnProcesar;
    JTextArea txtS;
    JScrollPane scpScroll;
```

```
// -----
// Crea la interfaz gráfica de usuario
public void init() {
    getContentPane().setLayout(null);

    btnProcesar = new JButton("Procesar");
    btnProcesar.setBounds(190, 15, 100, 23);
    btnProcesar.addActionListener(this);
    getContentPane().add(btnProcesar);

    txtS = new JTextArea();
    txtS.setFont(new Font("monospaced", 0, 12));

    scpScroll = new JScrollPane(txtS);
    scpScroll.setBounds(15, 53, 450, 150);
    getContentPane().add(scpScroll);
}
// -----
```

```
// Procesa eventos de tipo ActionEvent
public void actionPerformed( ActionEvent e ){
    if( e.getSource() == btnProcesar )
        procesar();
}
```

```
// -----
// Imprime una línea de texto con un salto de línea
void imprimir(String cad){
    txtS.append(cad + "\n");
}
// -----
```

```
// Método del botón Procesar
void procesar(){
```

```
    // Declaración e inicialización de variables locales
    int conta = 0, numero, unidades, millares;
```

```
// Borra la pantalla
txtS.setText("");

// Genera 200 números de 100 a 999 y cuenta cuantos son capicúas
for( int i = 0; i < 200; i++ ){

    // Genera un número aleatorio en el intervalo de 100 a 999
    numero = (int)(900*Math.random() + 100);

    // Muestra el número generado
    imprimir(numero + "");

    // Obtiene las cifras de las unidades y de los millares
    unidades = numero%10;
    millares = numero/1000;

    // Como el número tiene tres cifras, si la cifra de las unidades
    // es igual a la de los millares, entonces es capícua
    if( unidades == millares )
        conta++;
}

// Imprime la cantidad de capicúas
imprimir("Cantidad de números capicuas : " + conta);
}
```


Problemas propuestos

Actividad

1. Diseñe un programa que imprima la siguiente serie en una columna a razón de un término por fila.

1, 2, 3, 4, 5, 6, 7, 8, ..., 200

2. Diseñe un programa que imprima y sume 50 términos de la siguiente serie. Los términos serán mostrados en una columna a razón de un término por fila.

2, 4, 6, 8, 10, 12, 14, 16,

3. Diseñe un programa que imprima y sume 100 términos de la siguiente serie. Los términos serán mostrados en una columna a razón de un término por fila.

$\frac{2}{3}$, $\frac{5}{5}$, $\frac{8}{7}$, $\frac{11}{9}$, ...

4. Diseñe un programa que imprima y sume n términos de la siguiente serie. Los términos serán mostrados en una columna a razón de un término por fila

$\frac{1}{2}$, $\frac{4}{4}$, $\frac{7}{6}$, $\frac{10}{8}$, ...

5. Diseñe un programa que simule varios lanzamientos de un dado hasta obtener un seis. Muestre los puntajes del dado conforme se vayan generando y muestre al final cuántos lanzamientos fueron necesarios efectuar

6. Diseñe un programa que genere números aleatorios en el intervalo de -50 a 50 hasta obtener un número igual a -25 o igual a +25. Muestre los números conforme se vayan generando y muestre al final los números máximo y mínimo que fueron generados.

7. Diseñe un programa que genere números aleatorios en el intervalo de 100 a 900 hasta obtener un número par que cumpla con ser menor que 300 ó mayor que 700. Muestre los números generados, la suma de todos los números generados, el mayor número generado y el menor número generado.

8. Diseñe un programa que genere aleatoriamente las notas de una práctica calificada para 45 alumnos de una sección y determine la nota promedio de la sección y, las notas máxima y mínima generadas.

9. Diseñe un programa que genere aleatoriamente la nota de un alumno y muestre la nota seguido de tantos asteriscos como indique la nota.

Nota Asteriscos

10 *****

10. Diseñe un programa que genere aleatoriamente las notas de una práctica calificada de 30 alumnos de una sección e imprima por cada nota tantos asteriscos como indique la nota.

Nota	Histograma
10	*****
8	*****
5	*****
15	*****
4	****
1	*
20	*****
.	
.	
.	
7	*****
3	***

11. Diseñe un programa que determine la cantidad de divisores de un número natural.
12. Diseñe un programa que determine la cantidad de divisores de cada uno de los números del 2 al 100.
13. Diseñe un programa que genere números aleatorios en el intervalo de 1 a 1000 hasta obtener un número con cuatro divisores. El programa mostrará un listado como el siguiente:

Número	Cantidad de Divisores
121	3
881	2
60	12
978	8
964	6
22	4

14. Diseñe un programa que imprima los números del 1 al 100 a razón de 10 números por fila.

1	2	3	4	5	6	7	8	9	10
11	12	13	14	14	16	17	18	19	20
.....									
.....									
.....									
91	91	93	94	95	96	97	98	99	100

Autoevaluación

15. Diseñe un programa que halle la suma de todos los números enteros naturales múltiplos de 3 pero no de 5 que sean menores o iguales que n.

16. Diseñe un programa que halle la suma de n términos de la siguiente serie:

3, -8, 15, -24, ...

17. Diseñe un programa para calcular la suma de n términos de la siguiente serie:

1, - 1/2, 1/3, - 1/4, 1/5,

18. Diseñe un programa que halle la suma de n términos de la siguiente serie:

1, 6/5, 11/9, 16/13, ...

19. Diseñe un programa que imprima n términos de la siguiente serie. Note que a partir del tercer término, cada término es igual a la suma de los dos términos anteriores.

1, 2, 3, 5, 8, 13, 21, ...

20. Diseñe un programa que calcule el valor de π de acuerdo a la siguiente expresión:

$$\pi = 4 - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} - \frac{4}{11} + \dots (n \text{ términos})$$

21. Diseñe un programa que genere la tabla de multiplicar de un número entero n, desde n x 1 hasta n x 12. Por ejemplo, para n igual a 3 el programa imprimirá:

3	x	1	=	3
3	x	2	=	6
3	x	3	=	9
3	x	4	=	12
3	x	5	=	15
3	x	6	=	18
3	x	7	=	21
3	x	8	=	24
3	x	9	=	27
3	x	10	=	30
3	x	11	=	33
3	x	12	=	36

22. Diseñe un programa que imprima los primeros m múltiplos de un número entero positivo n.

23. Diseñe un programa para determinar el factorial de un número. El factorial de un número n se define como: $1 \times 2 \times 3 \times 4 \times 5 \times \dots \times (n-1) \times n$. El factorial de 0 y de 1 es 1.

24. Diseñe un programa que tenga determine la potencia m^n siendo m un número real y n un número entero. Considere que n puede ser positivo, negativo o cero. No use ningún método del lenguaje Java.

25. El movimiento de un proyectil viene dado por la siguiente ecuación:

$$h = 50 + 200t + 5t^2$$

Siendo h la altura que alcanza el proyectil en metros para un tiempo t en segundos. Diseñe un programa que imprima una tabla de valores de t vs h para valores del tiempo en el intervalo de 0 a 20 segundos cada 2 segundos.

26. Dado un número entero mayor que 1, diseñe un programa que determine el mayor divisor del número que sea diferente del número. Así, por ejemplo, dado el número 45, su mayor divisor distinto de 45 es 15.

27. Un número perfecto es un entero positivo, que es igual a la suma de todos sus divisores (excepto el mismo). El primer número perfecto es 6, ya que sus divisores son 1, 2, 3 y suman 6. Diseñe un programa que lea un entero positivo y determine si es o no es perfecto.

28. Diseñe un programa que determine si un número es primo. Un número es primo si tiene únicamente dos divisores: 1 y el mismo número. Por ejemplo, 3 es primo porque sus únicos divisores son 1 y 3; en cambio, 8 no es primo porque a parte del 1 y el 8 tiene otros divisores.

29. Diseñe un programa que genere números aleatorios en el intervalo de 10 a 99 hasta obtener un número cuya suma de sus cifras sea igual a 4 ó 6. Muestre los números conforme se vayan generando y muestre al final:

- La cantidad de números generados.
- El mayor número generado.
- El menor número generado.

30. Diseñe un programa que genere números aleatorios en el intervalo de -1000 a 1000 hasta obtener un número en el intervalo de -500 a 500 o hasta obtener 100 números, lo que ocurra primero. Muestre los números conforme se vayan generando y muestre al final:

- La cantidad de números generados.
- El mayor número positivo generado.
- El menor número positivo generado.
- El mayor número negativo generado.
- El menor número negativo generado.

31. Diseñe un programa que genere aleatoriamente los sueldos de 120 empleados de una empresa con valores en el intervalo de S/. 800 a S/. 1700 y determine:

- El sueldo promedio.
- El sueldo máximo.
- El sueldo mínimo.
- El número de empleados ganan un sueldo inferior a S/. 1000.

32. Diseñe un programa que genere una serie de números aleatorios en el intervalo de 1 a 10 hasta obtener dos números seguidos iguales e imprima cuantos números fueron necesarios generar.
33. Diseñe un programa que simule el lanzamiento simultáneo de tres dados hasta obtener 6 en los tres dados y determine cuantos lanzamientos simultáneos fueron necesarios efectuar.
34. Diseñe un programa que determine el revés de un número natural cuya cifra de las unidades es distinto de 0. Así, si se ingresa el número 1734, el programa imprimirá 4371.
35. Diseñe un programa que determine si todas las cifras de un número natural son o no consecutivas de izquierda a derecha o de derecha a izquierda. Así, por ejemplo: 1234 tiene todas sus cifras consecutivas de izquierda a derecha, 765432 tiene todas sus cifras consecutivas de derecha a izquierda; pero 82432 no tiene todas sus cifras consecutivas
36. Diseñe un programa que lea un número entero positivo n y determine cuantos números aleatorios deben generarse, en el intervalo de 1 a n , para obtener una suma mayor que 500.
37. Simule un juego en el cual dos jugadores tirarán los dados. El jugador A empieza con S/.15 y el jugador B con S/.23. Cuando hacen sus tiros, el jugador con la puntuación más alta gana S/.1. Continúan jugando hasta que uno de los dos pierda todo su dinero o hasta completar 100 juegos, lo que ocurra primero. Muestre un mensaje adecuado al final.
38. Diseñe un programa que determine la suma de las cifras de un número natural.
39. Diseñe un programa que muestre todos los números primos comprendidos en el intervalo de 1 a 100.
40. Diseñe un programa que muestre los 50 primeros números primos.
41. Diseñe un programa que muestre los 5 primeros números perfectos. El primer número perfecto es el 6. Como información, el 39 avo número perfecto tiene 4053496 cifras.
42. Diseñe un programa que imprima todos los puntos del primer cuadrante del plano cartesiano que cumplan con la inecuación: $x + y < 100$.
43. Diseñe un programa que imprima la siguiente tabla de valores:

N	$10^1 * N$	$10^2 * N$	$10^3 * N$	$10^4 * N$
1	10	100	1000	10000
2	20	200	2000	20000
3	30	300	3000	30000
4	40	400	4000	40000
5	50	500	5000	50000
6	60	600	6000	60000
7	70	700	7000	70000
8	80	800	8000	80000
9	90	900	9000	90000

44. Diseñe un programa que imprima n filas de la siguiente tabla:

3	5	7	9	11
4	6	8	10	12
5	7	9	11	13
6	8	10	12	14
.
.

Para recordar

- La estructura **while** verifica primero su condición de control antes ejecutar el cuerpo del bucle por lo que no se ejecutará nunca si a la primera vez la condición resulta falsa.
- La estructura **do...while** ejecuta primero el cuerpo del bucle, después de lo cual prueba su condición de control por lo que el cuerpo del bucle se ejecutará por lo menos una vez.
- La estructura **for** funciona como una estructura **while** controlada por contador, por lo que el cuerpo del bucle no se ejecutará nunca si la condición del **for** resulta falsa a la primera vez.
- Si la condición de control de una estructura repetitiva **do..while**, **while** o **for** nunca se vuelve falsa, se generará un bucle infinito.

ANEXO

1

PANELES

1. QUÉ ES UN PANEL

Un Panel es un objeto de la clase **JPanel** que se utiliza como contenedor de componentes visuales (JButton, JTextField, JLabel, JTextArea, etc) y de otros paneles. Los paneles se pueden utilizar para presentar pantallas diferentes dentro de un mismo applet o para dividir el applet en zonas.

Considere, por ejemplo, el caso de un applet que saluda a una persona por su nombre. El applet presentará dos paneles. En el primer panel se pedirá el nombre de la persona y en el segundo, se mostrará el saludo correspondiente. Para pasar del primer panel al segundo, se usará el botón **Siguiente** y para regresar del segundo panel al primero, se usará el botón **Anterior**.

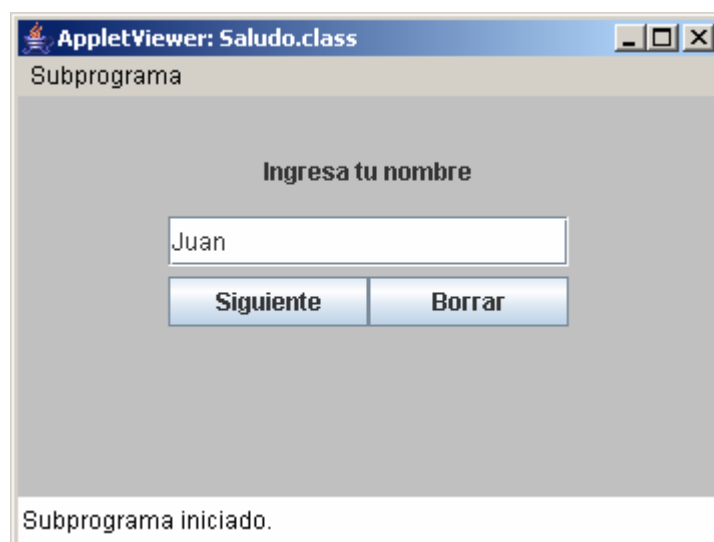


Figura 1 Applet Saludo mostrando el primer panel



Figura 2 Applet Saludo mostrando el segundo panel

No se trata de dos programas, sino de uno solo con dos paneles. Ambos paneles están contenidos dentro del applet, uno exactamente encima del otro. Al principio sólo es visible el panel de entrada. Al pulsar el botón Siguiente, se oculta el panel de entrada y se hace visible el panel de saludo. Al pulsar el botón Anterior, se oculta el panel de salida y se hace visible el panel de entrada. En este caso el panel de contenido del applet contiene únicamente a los paneles.

2. CREACIÓN DE UN PANEL

Para crear un panel se siguen los siguientes pasos:

Primero: Declarar una variable referencia de tipo JPanel

```
// Esto declara una variable pnlEntrada de tipo JPanel
Panel pnlEntrada;
```

Segundo: En el método init, crear el objeto JPanel y fijar sus atributos

```
// Esto crea el objeto JPanel.
pnlEntrada = new JPanel();

// Esto anula el posicionamiento automático de componentes dentro
// del panel, así podremos ubicar elementos en el panel mediante
// coordenadas.
pnlEntrada.setLayout(null);

// Esto fija la ubicación y las dimensiones del panel. Para que quepa
// exactamente en toda la extensión del applet, las coordenadas deben
// ser (0,0) y, el ancho y el alto deben ser iguales a los del applet.
pnlEntrada.setBounds(0,0,350,200);

// Esto fija la visibilidad del panel. Si está trabajando con
```

```
// paneles que simulan ventanas, asegúrese de que sólo un panel
// se encuentre visible en un momento dado. Al principio se fija
// en true la visibilidad del panel que se mostrará cuando cargue
// el programa. La visibilidad de los demás paneles deben
// estar en false.
pnlEntrada.setVisible(true);

// Esto fija el color del fondo del panel (opcional).
pnlEntrada.setBackground(Color.lightGray);
```

Tercero : Adicionar el panel al applet

```
// Esto adiciona el panel al panel de contenido del applet
getContentPane().add(pnlEntrada);
```

3. ADICIÓN DE UN COMPONENTE VISUAL A UN PANEL

Para agregar un componente visual, previamente creado, a un panel, se usa la siguiente instrucción:

```
// Esto adiciona el botón btnSiguiente al panel pnlEntrada. Para esto
// el botón y el panel ya tienen que haber sido creados.
pnlEntrada.add(btnSiguiente);
```

Observe que para adicionar un componente a un Panel también se usa el método **add**, pero precedido del nombre del panel al que se añadirá el componente.

4. OCULTAR / MOSTRAR UN PANEL

Al trabajar con paneles que simulan ventanas, sólo será visible un panel en un momento dado. Para poder pasar de un panel a otro, se requiere ocultar un panel y mostrar el otro. Esto se lleva a cabo utilizando el método **setVisible** como se muestra a continuación.

```
// Esto hace visible el panel pnlSalida
// Para esto se usa el valor true dentro de setVisible
pnlSalida.setVisible(true);

// Esto oculta el panel pnlEntrada
// Para esto se usa el valor false dentro de setVisible
pnlEntrada.setVisible(false);
```

5. CODIGO FUENTE DEL EJEMPLO

```
import java.awt.event.*;
import java.awt.*;
import javax.swing.*;

public class Saludo extends JApplet implements ActionListener{
    // Declaración de variables globales
```

```

JPanel pnlEntrada, pnlSalida;
JLabel lblNombre, lblSaludo;
JButton btnSiguiente, btnBorrar, btnAnterior;
JTextField txtNombre;

// -----
// Crea la GUI
public void init() {
    getContentPane().setLayout(null);
    crearPanelEntrada();
    crearPanelSalida();
}
// -----
// Crea el panel pnlEntrada
void crearPanelEntrada() {

    pnlEntrada = new JPanel();
    pnlEntrada.setLayout(null);
    pnlEntrada.setBounds(0,0,350,200);
    pnlEntrada.setVisible(true);
    pnlEntrada.setBackground(Color.lightGray);
    getContentPane().add(pnlEntrada);

    lblNombre = new JLabel("Ingresa tu nombre", JLabel.CENTER);
    lblNombre.setBounds(0,25,350,25);
    pnlEntrada.add(lblNombre);

    txtNombre = new JTextField();
    txtNombre.setBounds(75,60,200,25);
    pnlEntrada.add(txtNombre);

    btnSiguiente = new JButton("Siguiente");
    btnSiguiente.setBounds(75,90,100,25);
    btnSiguiente.addActionListener(this);
    pnlEntrada.add(btnSiguiente);

    btnBorrar = new JButton("Borrar");
    btnBorrar.setBounds(175,90,100,25);
    btnBorrar.addActionListener(this);
    pnlEntrada.add(btnBorrar);
}
// -----
// Crea el panel pnlSalida
void crearPanelSalida() {

    pnlSalida = new JPanel();
    pnlSalida.setLayout(null);
    pnlSalida.setBounds(0,0,350,200);
    pnlSalida.setVisible(false);
    pnlSalida.setBackground(Color.gray);
    getContentPane().add(pnlSalida);

    lblSaludo = new JLabel("", JLabel.CENTER);
    lblSaludo.setFont(new Font("dialog", Font.BOLD, 16));
    lblSaludo.setBounds(0,50,350,25);
    pnlSalida.add(lblSaludo);
}

```

```
        btnAnterior = new JButton("Anterior");
        btnAnterior.setBounds(125,150,100,25);
        btnAnterior.addActionListener(this);
        pnlSalida.add(btnAnterior);
    }
    // -----
    // Procesa eventos de tipo(ActionEvent)
    public void actionPerformed( ActionEvent e ){

        if( e.getSource() == btnSiguiente )
            siguiente();
        if( e.getSource() == btnAnterior )
            anterior();
        if( e.getSource() == btnBorrar )
            borrar();
    }
    // -----
    // Método para el botón btnSiguiente
    void siguiente(){

        // Declaración de variables locales
        String nombre;

        // Ingreso del nombre
        nombre = txtNombre.getText();

        // Impresión del saludo
        lblSaludo.setText("Hola " + nombre);

        // Configuración de paneles
        pnlSalida.setVisible(true);
        pnlEntrada.setVisible(false);
    }
    // -----
    // Método para el botón btnAnterior
    void anterior(){

        // Configuración de paneles
        pnlSalida.setVisible(false);
        pnlEntrada.setVisible(true);

        // Pone el cursor en txtNombre y selecciona su contenido
        txtNombre.requestFocus();
        txtNombre.selectAll();
    }
    // -----
    // Método para el botón btnBorrar
    void borrar(){
        txtNombre.setText("");
        txtNombre.requestFocus();
    }
}
```


ANEXO

2

CLASES BÁSICAS DE SWING

CONTENIDO

1. Clase JLabel
2. Clase JButton
3. Clase JTextField
4. Clase JTextArea
5. Clase JComboBox
6. Clase JScrollPane
7. Métodos comunes a todas las clases
8. Obtención de datos de un TextField
9. Concatenación de cadenas
10. Conversión de un número a cadena

En este anexo se muestran únicamente los componentes de Swing que usaremos en el curso. Se muestran también, por cada componente, sus métodos principales. Revise la documentación del lenguaje para obtener mayor información.

1. Clase JLabel

Un objeto JLabel es un componente utilizado para mostrar una línea de texto fijo y/o una imagen fija.

Declaración de una referencia de tipo JLabel

- ♦ Declara el nombre de una referencia de tipo JLabel

```
JLabel nombre;
```

Constructores

- ♦ Crea un objeto JLabel sin imagen y con texto vacío.

```
nombre = new JLabel();
```

- ♦ Crea un objeto JLabel con el texto especificado que será alineado horizontalmente a la izquierda y centrado verticalmente. La alineación horizontal y vertical es dentro de su área de visualización.

```
nombre = new JLabel("texto");
```

Donde:

texto es el texto a ser mostrado.

- ♦ Crea un objeto JLabel con el texto y la alineación horizontal especificados. El texto será centrado verticalmente. La alineación horizontal y vertical es dentro de su área de visualización.

```
nombre = new JLabel("texto", alineacion);
```

Donde:

texto es el texto a ser mostrado.

alineacion es la alineación horizontal y puede ser uno de los siguientes:

SwingConstants.CENTER	centrado horizontalmente
SwingConstants.RIGHT	alineado a la derecha
SwingConstants.LEFT	alineado a la izquierda

- ♦ Crea un objeto JLabel con la imagen especificada centrada horizontal y verticalmente. La alineación horizontal y vertical es dentro de su área de visualización.

```
nombre = new JLabel(new ImageIcon("imagen")) ;
```

Donde:

imagen es el nombre del archivo imagen que puede ser gif o jpg.

- ♦ Crea un objeto JLabel con la imagen y alineación horizontal especificadas. La imagen será centrada verticalmente. La alineación horizontal y vertical es dentro de su área de visualización.

```
nombre = new JLabel(new ImageIcon("imagen"), alineacion) ;
```

Donde:

imagen es el nombre del archivo imagen que puede ser gif o jpg.

alineacion es la alineación horizontal y puede ser uno de los siguientes:

SwingConstants.CENTER	centrado horizontalmente
SwingConstants.RIGHT	alineado a la derecha
SwingConstants.LEFT	alineado a la izquierda

Métodos

- ♦ Fija el texto del JLabel con el texto especificado.

```
nombre.setText("texto") ;
```

Donde:

texto es el texto a ser mostrado.

- ♦ Fija la imagen del JLabel con la imagen especificada.

```
nombre.setIcon("imagen") ;
```

Donde:

imagen es el nombre del archivo imagen que puede ser gif o jpg.

- ♦ Fija la alineación horizontal del texto y la imagen del JLabel, en conjunto, dentro del área de visualización.

```
nombre.setHorizontalAlignment(alineacion) ;
```

Donde:

alineacion es la alineación horizontal y puede ser uno de los siguientes:

SwingConstants.CENTER centrado horizontalmente
 SwingConstants.RIGHT alineado a la derecha
 SwingConstants.LEFT alineado a la izquierda

- ♦ Fija la alineación vertical del texto y la imagen del JLabel, en conjunto, dentro del área de visualización.

```
nombre.setVerticalAlignment(alineacion);
```

Donde:

alineacion es la alineación vertical y puede ser uno de los siguientes:

SwingConstants.CENTER centrado verticalmente
 SwingConstants.TOP alineado hacia la parte superior
 SwingConstants.BOTTOM alineado hacia la parte inferior

2. Clase JButton

Permite crear botones de pulsación.

Declaración de una referencia de tipo JButton

- ♦ Declara el nombre de una referencia de tipo JButton.

```
JButton nombre;
```

Constructores

- ♦ Crea un objeto JButton sin imagen y sin texto.

```
nombre = new JButton();
```

- ♦ Crea un objeto JButton con el texto especificado. El texto queda centrado horizontal y verticalmente dentro de su área de visualización.

```
nombre = new JButton("texto");
```

Donde:

texto es el texto a ser mostrado.

- ♦ Crea un objeto JButton con la imagen especificada. La imagen queda centrada horizontal y verticalmente dentro de su área de visualización.

```
nombre = new JButton(new ImageIcon("imagen"));
```

Donde:

imagen es el nombre del archivo imagen que puede ser gif o jpg.

- ♦ Crea un objeto JButton con la imagen y el texto especificados. La imagen y el texto en conjunto quedan centrados horizontal y verticalmente dentro de su área de visualización.

```
nombre = new JButton("texto", new ImageIcon("imagen"));
```

Donde:

texto es el texto a ser mostrado.

imagen es el nombre del archivo imagen que puede ser gif o jpg.

Métodos

- ♦ Fija el texto del JButton con el texto especificado.

```
nombre.setText("texto");
```

Donde:

texto es el texto a ser mostrado.

3. Clase JTextField

Un objeto JTextField es un componente utilizado para la entrada de datos en forma de texto compuesto de una sola línea.

Declaración de una referencia de tipo JTextField

- ♦ Declara el nombre de una referencia de tipo JTextField.

```
JTextField nombre;
```

Constructores

- ♦ Crea un objeto JTextField con texto vacío.

```
nombre = new JTextField();
```

- ♦ Crea un objeto JTextField con el texto especificado. El texto queda alineado horizontalmente hacia la izquierda y centrado verticalmente dentro de su área de visualización.

```
nombre = new JTextField("texto");
```

Donde:

texto es el texto a ser mostrado.

Métodos

- ♦ Obtiene el texto del JTextField y lo almacena en una variable.

```
variable = nombre.getText();
```

Donde:

variable debe ser de tipo **String**.

- ♦ Fija el texto del JTextField por el texto especificado.

```
nombre.setText("texto");
```

Donde:

texto es el texto a ser mostrado.

- ♦ Fija el **indicador** que determina si el texto del JTextField es o no editable.

```
nombre.setEditable(indicador);
```

Donde **indicador** puede ser:

true Permite que el contenido del JTextField sea editable.

false Impide que el contenido del JTextField sea editable.

- ♦ Fija la alineación horizontal del texto del JTextField.

```
nombre.setHorizontalAlignment(alineacion);
```

Donde:

alineacion es la alineación horizontal y puede ser uno de los siguientes:

SwingConstants.CENTER centrado horizontalmente

SwingConstants.RIGHT alineado a la derecha

SwingConstants.LEFT alineado a la izquierda

4. Clase JTextArea

Un objeto JTextArea presenta un área de texto en una zona rectangular que no posee barras de desplazamiento. Para poder visualizar el texto que se encuentra más allá de la zona rectangular se requiere de un objeto JScrollPane.

Declaración de una referencia de tipo JTextArea

- ♦ Declara el nombre de una referencia de tipo JTextArea.

```
JTextArea nombre;
```

Constructores

- ♦ Crea un objeto JTextArea.

```
nombre = new JTextArea();
```

- ♦ Crea un objeto JTextArea con el texto especificado.

```
nombre = new JTextArea("texto");
```

Métodos

- ♦ Obtiene el texto del JTextArea y lo almacena en una variable.

```
variable = nombre.getText();
```

Donde:

variable debe ser declarada de tipo **String**.

- ♦ Fija el texto del JTextArea con el texto especificado. Esto reemplaza el contenido actual del JTextArea.

```
nombre.setText("texto");
```

Donde:

texto es el texto a ser mostrado.

- ♦ Añade un nuevo texto al final del texto actual del objeto JTextArea.

```
nombre.append("texto");
```

Donde:

texto es el texto a ser añadido.

- ♦ Fija el **indicador** que determina si el texto del JTextArea es o no editable.

```
nombre.setEditable(indicador);
```

Donde **indicador** puede ser:

- true Permite que el contenido del JTextField sea editable.
- false Impide que el contenido del JTextField sea editable.

5. Clase JComboBox

Declaración de una referencia de tipo JComboBox

- ♦ Declara el nombre de una referencia de tipo JComboBox.

```
JComboBox nombre;
```

Constructores

- ♦ Crea un nuevo objeto JComboBox vacío.

```
nombre = new JComboBox();
```

Métodos propios

- ♦ Adiciona el ítem especificado al JComboBox. El ítem se adiciona al final de los ítems existentes.

```
nombre.addItem("item");
```

Donde:

item es el ítem a ser adicionado.

- ♦ Obtiene el índice del ítem seleccionado y lo almacena en una variable. El primer ítem tiene índice 0, el segundo índice 1 y así sucesivamente.

```
indice = nombre.getSelectedIndex();
```

Donde:

variable debe ser declarada de tipo **int**.

- ♦ Obtiene el texto del ítem seleccionado y lo almacena en una variable.

```
item = (String)nombre.getSelectedItem();
```

Donde:

variable debe ser declarada de tipo **String**.

- ♦ Obtiene el texto de un ítem dado su índice y lo almacena en una variable.

```
variable = (String)nombre.getItemAt(indice);
```

Donde:

variable es la variable que recibe el ítem. Debe ser de tipo **String**.

índice es el índice del ítem obtenido.

- ♦ Selecciona un ítem dado su índice.

```
nombre.setSelectedIndex(indice);
```

Donde :

índice es el índice del ítem a seleccionar.

6. Clase JScrollPane

Un objeto JScrollPane proporciona una vista con barras de desplazamiento para otros objetos como es el caso de un objeto JTextArea.

Declaración de una referencia de tipo JScrollPane

- ♦ Declara el nombre de una referencia de tipo JScrollPane.

```
JScrollPane nombre;
```

Constructores

- ♦ Crea un objeto JScrollPane que muestra el contenido del JTextArea especificado.

```
nombre = new JScrollPane(nombreTextArea);
```

Donde:

nombreTextArea es el nombre del TextArea a ser recorrido.

7. Métodos comunes a todas las clases

- ♦ Fija la ubicación y las dimensiones del componente.

```
nombre.setBounds(x1, y1, ancho, alto);
```

Donde:

x1	:	coordenada x de la esquina superior izquierda
y1	:	coordenada y de la esquina superior izquierda
ancho	:	ancho de la zona rectangular del Label
alto	:	altura de la zona rectangular del Label

- ♦ Fija el color del texto del componente con el color especificado.

Usando colores predefinidos

```
nombre.setForeground(Color.nombrecolor);
```

Donde:

nombrecolor puede ser uno de los siguientes:

orange
pink
cyan
magenta
yellow
black
white
gray
lightGray
darkGray
red
green
blue

Usando colores creados

Se crea combinando cantidades de rojo, verde y azul. A estos colores se conocen como colores RGB: R de red (rojo), G de green (verde) y B de blue (azul).

```
nombre.setForeground(new Color(rojo, verde, azul));
```

Donde:

rojo cantidad de rojo. Puede ser un entero de 0 a 255.

verde cantidad de verde. Puede ser un entero de 0 a 255.

azul cantidad de azul. Puede ser un entero de 0 a 255.

- ♦ Fija el color del fondo del componente con el color especificado.

```
nombre.setBackground(color);
```

Donde **color** es el mismo que el usado por `setForeground`.

- ♦ Fija la visibilidad del componente.

```
nombre.setVisible(indicador);
```


Donde **indicador** puede ser:

true el componente es visible
false el componente es invisible

- ♦ Habilita o deshabilita el componente.

```
nombre.setEnabled(indicador);
```

Donde **indicador** puede ser:

true habilita el componente
false deshabilita el componente

- ♦ Fija la fuente del componente.

```
nombre.setFont(new Font(fuente, estilo, tamaño));
```

Donde:

fuente : nombre de la fuente entre comillas. Puede ser:

dialoginput, monospaced, serif, sansserif, dialog

estilo : estilo de la fuente. Puede ser:

Font.BOLD → Negrita
Font.ITALIC → Cursiva
Font.PLAIN → Plano

Los estilos pueden sumarse:

Font.BOLD+Font.ITALIC → Negrita + Cursiva
Font.PLAIN+Font.BOLD → Plano + Negrita
Font.ITALIC+Font.PLAIN → Plano + Cursiva

tamaño : Tamaño de la fuente en puntos.

- ♦ Otorga el foco a un componente.

```
nombre.requestFocus();
```

Donde:

nombre es el nombre del componente que recibe el foco.

- ♦ Fija la opacidad de un componente.

```
nombre.setOpaque(indicador);
```

Donde:

indicador puede ser uno de los siguientes:

true el componente es opaco

false el componente es transparente

Todos los componentes son opacos por defecto a excepción del JLabel que es transparente. Para pintar el fondo del JLabel hay que fijarlo como opaco.

8. Obtención de datos de un TextField

En los casos que siguen asumimos que el nombre del TextField desde donde se quiere obtener el dato es **txtDato**.

Leer una cadena

```
variable = txtDato.getText();
```

Leer un entero

```
variable = Integer.parseInt( txtDato.getText() );
```

Leer un real

```
variable = Double.parseDouble( txtDato.getText() );
```

Leer un caracter

```
variable = txtDato.getText().charAt(0);
```

9. Concatenación de cadenas

El operador de concatenación de cadenas + permite unir dos cadenas. Si uno de los operandos no es cadena, es convertido automáticamente a cadena antes de efectuar la concatenación.

Ejemplo

```
"Hola " + "mundo"      → "Hola mundo"
"Mi edad es " + 21      → "Mi edad es " + "21" → "Mi edad es 21"
"123" + 5               → "123" + "5"       → "1235"
"" + 125                → "" + "125"        → "125"
```

Ejemplo

Suponga que tiene una variable **edad** de tipo **int** cuyo valor es 25, entonces:

```
"Mi edad es " + edad
```

Produce el siguiente trámite:

```
"Mi edad es " + 25
```

```
"Mi edad es " + "25"
```

```
"Mi edad es 25"
```

10. Conversión de un número a cadena

Para convertir un número a cadena simplemente hay que sumar al número una cadena vacía en la forma:

```
" " + numero
```

o en la forma:

```
numero + " "
```

Ejemplo

Para convertir el número 2456 podemos escribir:

```
2456 + " " → "2456"
```

o también:

```
" " + 2456 → "2456"
```