

Estructuras de selección y repetición. En Java 2

- Operadores Aritméticos
- Operadores relacionales
- Condicionales simple if, y doble: if else
- Multi condicional : **Switch**
- Estructura de repetición **While y For**

Operadores Aritméticos

- Las expresiones aritméticas en Java deben escribirse en *forma de línea recta* para facilitar la introducción de programas en la computadora.
 - **Ejemplo:** "a dividido entre b" se deben escribir como a / b de modo que todas las constantes, variables y operadores aparezcan en una línea recta.
- Usamos paréntesis en las expresiones Java de la misma forma que en las expresiones algebraicas.
 - Por ejemplo, si queremos multiplicar a por la cantidad $b + c$, escribimos: $a * (b+c)$

Operadores Aritméticos

Operación de Java	Operador Aritmético	Expresión Algebraica	Expresión en Java
Suma	+	$f + 7$	$f+7$
Resta	-	$p - c$	$p-c$
Multiplicación	*	bm	$b * m$
División	/	$x \div y$, $\frac{x}{y}$	x/y
Residuo	%	$r \text{ mod } s$	$r \% s$

Precedencia de Operadores Aritméticos

Operadores	Operadores	Orden de evaluación (precedencia)
()	Paréntesis	<p>Se evalúan primero. Si los paréntesis están anidados, la expresión dentro del par más interno se evalúa primero.</p> <p>Si hay varios pares de paréntesis "en el mismo nivel" (es decir, no anidados), se evalúan de izquierda a derecha.</p>
*, /, %	Multiplicación, división, Residuo	Se evalúan en segundo lugar. Si hay varios, se evalúan izquierda a derecha
+, -	Suma y Resta	Se evalúan a1 final. Si hay varios, se evalúan de izquierda a derecha.

Ejemplos:

Ejemplo 1:

- Algebra: $y = mx + b$
- Java: $y = m * x + b;$

Ejemplo 2:

Algebra: $z = pr \% q + w/x - y$

Java: $p * r \% q + w/x - y;$

(1) (2) (4) (3) (5)

Toma de decisiones: Operadores de igualdad y relacionales

Operador de igualdad o relacional algebraico	Operador de igualdad o relacional en Java	Ejemplo en Java	Significado
Operadores de Igualdad			
=	==	x==y	x es igual a y
≠	!=	x != y	x no es igual a y
Operadores relacionales			
>	>	x >y	x es mayor que y
<	<	x<y	x es menor que y
≥	>=	x >= y	x es mayor o igual que y
≤	<=	x <= y	x es menor o igual que y

Uso del condicional If en Java

- Ejemplo:

```
// Empleo de enunciados if, operadores relacionales y operadores de igualdad
```

```
import java.awt.*;
```

```
import java.applet.Applet;
```

```
public class Comparison extends Applet {
```

```
Label prompt1;
```

```
TextField input1;
```

```
Label prompt2;
```

```
TextField input2;
```

```
int num1, num2;
```

Uso del condicional If en Java

// establecer componentes de interfaz gráfica con el usuario
e inicializar variables

```
public void init()
{
prompt1 = new Label ( "Teclee un entero" );
input1 = new TextField( 10 );
prompt2 = new Label( "Teclee un entero y pulse Enter: ");
input2 = new TextField( 10 );
add( prompt1 ); // colocar prompt1 en la applet
add( input1 ); // colocar input1 en la applet
add( prompt2 ); // colocar prompt2 en la applet
add( input2 ); // colocar input2 en la applet
}
```

Uso del condicional If en Java

// exhibir los resultados

```
public void paint( Graphics g )
{
g.drawString( "Los resultados de la comparación son:", 70, 75 );
if ( num1 == num2 )
g.drawString( num1 + " == " + num2, 100, 90);
if ( num1 != num2 )
g.drawString( num1 + " != " + num2, 100, 105);
if ( num1 < num2 )
g.drawString( num1 + " < " + num2, 100, 120 );
if ( num1 > num2 )
g.drawString( num1 + " > " + num2, 100, 135 );
if ( num1 <= num2 )
g.drawString( num1 + " <= " + num2, 100, 150 );
if ( num1 >= num2 )
g.drawString( num1 + " >= " + num2, 100, 165 );
}
```

Uso del condicional If en Java

```
// procesar la acción del usuario en el campo de texto input2
public boolean action( Event event, Object o )
{
    if ( event.target == input2 )
    {
        num1 = Integer.parseInt( input1.getText() );
        num2 = Integer.parseInt( input2.getText() );
        repaint () ;
    }

    return true; // indica que se procesó la acción del usuario
}
```

La salida del programa anterior es la siguiente:

AppletViewer: hola/Hola.class

Subprograma

Teclee un entero

Teclee un entero y pulse Enter:

Los resultados de la comparación son:

1 != 2
1 < 2
1 <= 2

Subprograma iniciado.

AppletViewer: hola/Hola.class

Subprograma

Teclee un entero

Teclee un entero y pulse Enter:

Los resultados de la comparación son:

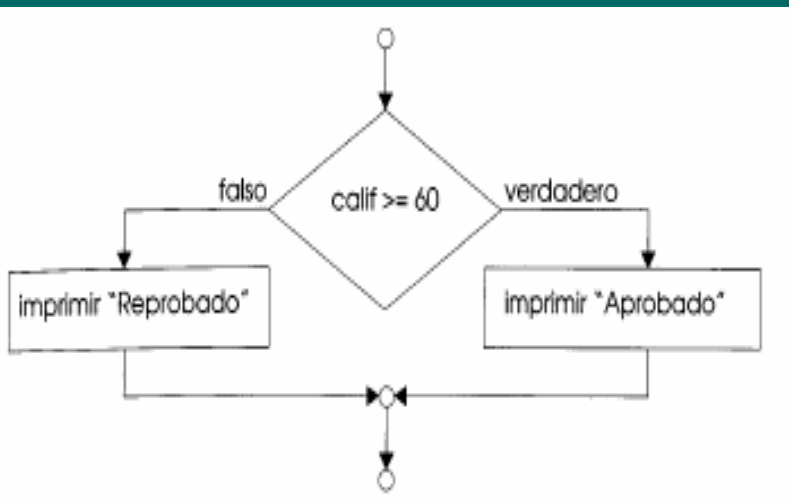
10 != 15
10 < 15
10 <= 15

Subprograma iniciado.

Estructura if else:

Ejemplo: El siguiente segmento de programa imprimirá “Aprobado” si un estudiante obtiene una calificación mayor o igual a 60, en caso contrario “Reprobado”

Diagrama de Flujo



Código Java

```
if ( calif >= 60 )  
System.out.println( "Aprobado" )  
else  
System.out.println( "Reprobado" );
```

Estructura if else:

Otra forma de representar a la estructura if / else
haciendo uso del operador ?:

```
system.out.println( calif >= 60 ? "Aprobado" : "reprobado" );
```



Estructura if else Anidada:

Prueban múltiples casos colocando estructuras if / else dentro de estructuras if / else.

Ejemplo1: Imprimir **A** para las calificaciones de examen mayores o iguales que 90, **B** para las calificaciones en el intervalo de 80 a 89, **C** para las calificaciones en el intervalo de 70 a 79, **D** para las calificaciones en el intervalo de 60 a 69 y **F** para todas las demás calificaciones.

En Pseudocódigo

If calificación del estudiante \geq que 90

Imprimir "A"

else

If calificación del estudiante \geq 80

Imprimir "B"

else

If calificación del estudiante \geq que 70

Imprimir "C"

else

If calificación del estudiante \geq que 60

Imprimir "D"

else

Imprimir "F"

En Java

```
if ( calif >= 90 )
    System.out.println( "A" );
else if ( calif >= 80 )
    System.out.println( "B" );
else if ( calif >= 70 )
    System.out.println( "C" );
else if ( calif >= 60 )
    System.out.println( "D" );
else
    System.out.println ( "F" );
```

Multi condicional: Switch

Prueba individualmente una variable para cada uno de los valores enteros constantes que puede asumir, emprendiéndose en cada caso acciones distintas.

Ejemplo: El siguiente programa cuenta el numero de cada una de las diferentes calificaciones de letra que ciertos estudiantes obtuvieron en un examen.

```
// Recuento de calificaciones de letra
import java.awt.*;
import java.applet.Applet;
```

Multi condicional: Switch

EJEMPLO: Recuento de calificaciones

```
public class SwitchTest extends Applet {
Label prompt;
TextField input;
int aCount = 0, bCount = 0, cCount = 0, dCount = 0, eCount = 0;

public void init ()
{
prompt = new Label( "Teclee calificaciónn );
input = new TextField( 2 );
add( prompt );
add( input );
}
```

Multi condicional: Switch

EJEMPLO: Recuento de calificaciones

```
public void paint( Graphics g )
{
  g.drawString( "Totales para cada calificación de Letra: ",
    25, 40 );
  g.drawString( "A: " + aCount, 25, 55 );
  g.drawString( "B: " + bCount, 25, 70 );
  g.drawString( "C: " + cCount, 25, 85 );
  g.drawString( "D: " + dCount, 25, 100 );
  g.drawString( "F: " + fCount, 25, 115 );
}
```

```
public boolean action( Event e, Object o )
{
String val = o.toString();
char grade = val.charAt ( 0 );
showStatus( "" );
input.setText ( "" );

switch (grade) {

    case 'A' : case 'a':
        ++aCount ;
        break;

    case 'B': case 'b'
        ++bCount;
        break;

    case 'C': case 'c'
        ++cCount;
        break;
```

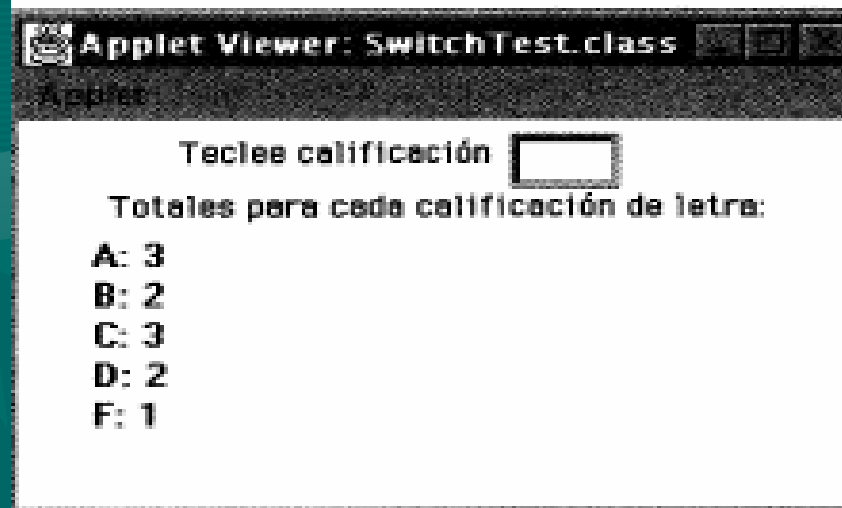
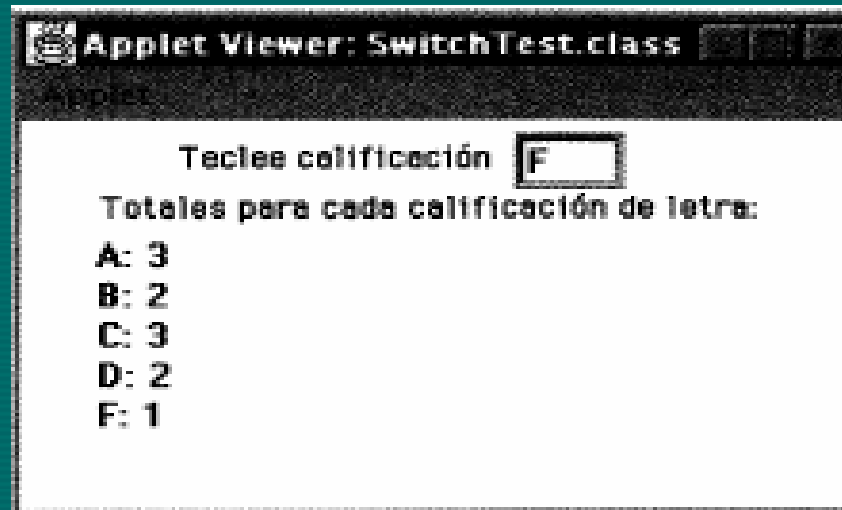
```
case 'D': case 'd'  
++dCount;  
break;
```

```
case 'F': case 'f'  
++fCount;  
break;
```

```
default :  
showStatus( " Calificación incorrecta. Teclee nueva  
calificación );  
break;
```

```
}  
repaint();  
return true;  
  
}  
}
```

Resultado de Ejemplo en Java



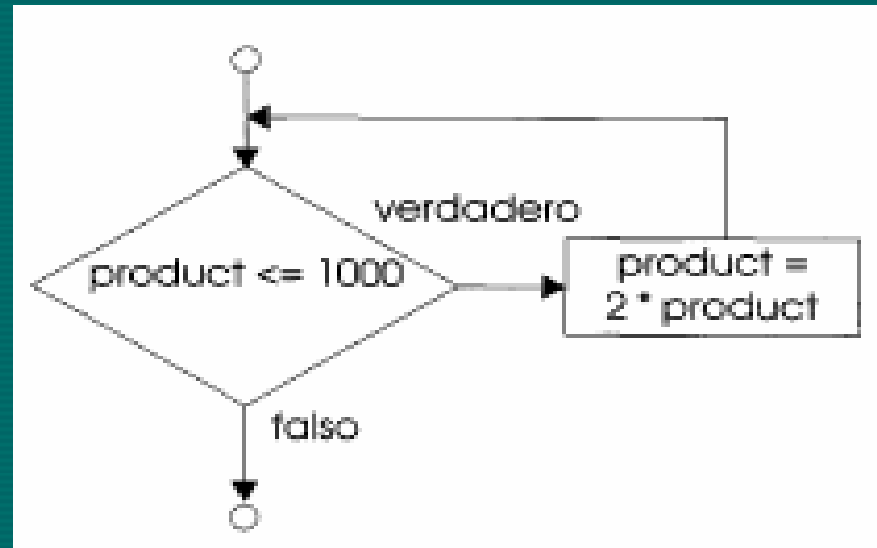
Estructura **While**:

Una *estructura de repetición* permite al programador especificar que una acción se repita en tanto se cumpla una condición.

Como ejemplo de `while`, considere un segmento de programa diseñado para encontrar la primera potencia de 2 mayor que 1000. Suponga que la variable entera **product** se ha inicializado en 2.

```
int product = 2;  
while ( product <= 1000 )  
product = 2 * product;
```

Diagrama de flujo : While



La variable `product` se multiplica repetidamente por 2, asumiendo los valores 4, 8, 16, 32, 64, 128, 256, 512 y 1024 sucesivamente. Cuando `product` se convierte en 1024, la condición de la estructura `while`, `product <= 1000` se vuelve falsa.

En Java:

El segmento de programa anterior escrito de forma completa en JAVA es el siguiente:

Ejemplo: Como ejemplo escribiremos un programa que calcule la primer potencia mayor que 1000 de un número cualquiera introducida por teclado.

```
// Búsqueda de la primer potencia de un numero mayor que  
1000
```

```
import java.awt.*;
```

```
import java.applet.Applet;
```

Búsqueda de la primer potencia de un numero mayor que 1000

```
public class potencia extends Applet {
Label prompt;
TextField input;
Int Numero;
Int Potencia = 1;

public void init ()
{
prompt = new Label( "Teclee el número del 1-99: " );
input = new TextField( 2 );
add( prompt );
add( input );
}
```

Búsqueda de la primer potencia de un numero mayor que 1000

```
// exhibir los resultados
public void paint( Graphics g )
{
g.drawString( "La primer potencia de " + Numero + " Mayor
que 1000 es: " + Potencia , 70, 75 )
}
```

Búsqueda de la primer potencia de un numero mayor que 1000

// procesar la acción del usuario

```
public boolean action( Event event, Object o )
```

```
{
```

```
    Numero = Integer.parseInt( input.getText() );
```

```
    While ( Potencia <= 1000 )
```

```
    {
```

```
        Potencia = Numero * Potencia;
```

```
    }
```

```
        repaint () ;
```

```
    return true;
```

```
}
```

```
}
```

Ciclo For:

Repetición controlada por contador

// Repetición controlada Por contador con estructura for

```
import java.awt .Graphics;
```

```
import java.applet.Applet;
```

```
public class Forcounter extends Applet {
```

```
public void paint( Graphics g )
```

```
{
```

```
int ypos = 25;
```

```
for ( int counter = 1; counter <= 10; counter++ )
```

```
{
```

```
g.drawString( Integer.toString( counter ), 25, yPos );
```

```
ypos += 15;
```

```
}
```

```
}
```

```
}
```

Ciclo For

El formato general de la estructura for es:

```
for ( expresión1; expresión2; expresión3 )  
Enunciado
```

Donde:

Expresión1 : Inicializa la variable de control.

Expresión2: Condición dentro del ciclo.

Expresión3: Incrementa la variable de control.

Su equivalente en estructura WHILE

Formato While

Expresión1;

while (Expresión2) {

Expresión3;

}

Ejemplo:

Suma = 0;

While (Suma<=100)

{

Suma = Suma + 2;

}

Diagrama de flujo de una estructura FOR

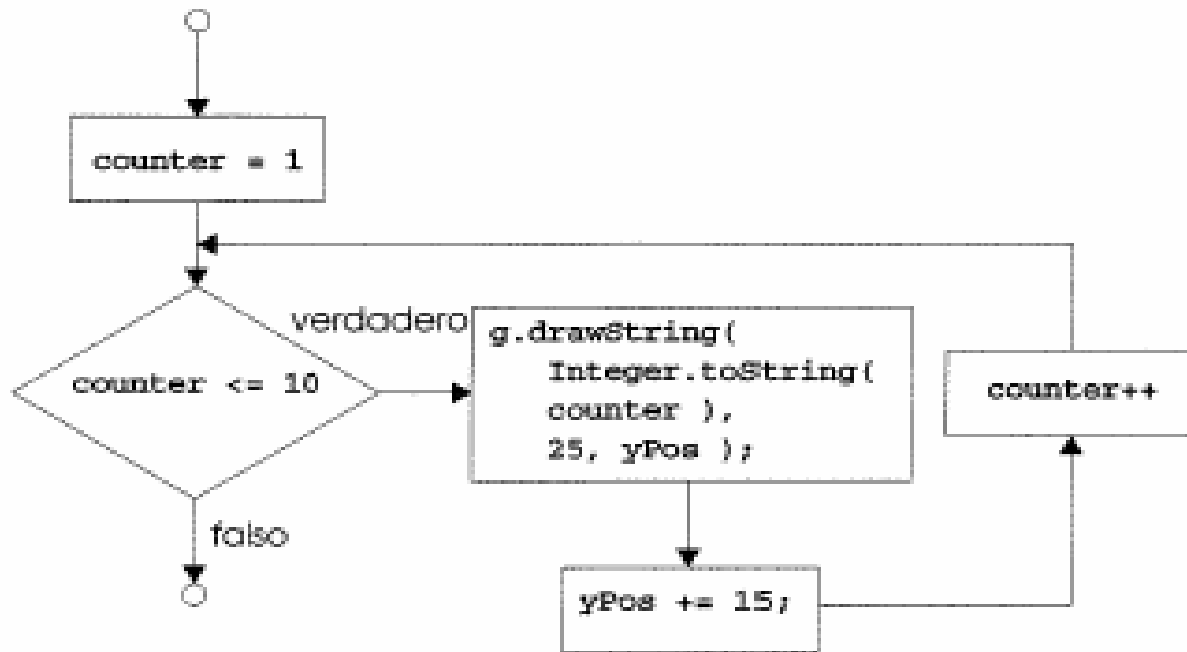


Diagrama de flujo de una estructura de repetición `for` típica.

Ejemplos de uso de la estructura For:

a) Variar la variable de control de 1 a 100 en incrementos de 1.

```
for ( int i = 1; i <= 100; i++ )
```

b) Variar la variable de control de 100 a 1 en incrementos de -1 (decrementos de 1).

```
for ( int i = 100; i >= 1; i-- )
```

c) Variar la variable de control desde 7 hasta 77 en incrementos de 7.

```
for ( int i = 7; i <= 77; i += 7 )
```

Ejemplos de uso de la estructura

For:

d) Variar la variable de control de 20 a 2 en pasos de - 2.

```
for ( int i = 20; i >= 2; i -= 2 )
```

e) variar la variable de control siguiendo esta secuencia de valores
2, 5, 8, 11, 14, 17, 20.

```
For( int j = 2; j<=20; j +=3)
```

f) Variar la variable de control siguiendo esta secuencia de valores
99, 88, 77, 66, 55, 44, 33, 22, 11, 0.

```
For( int j = 99; j>=0; j -=11)
```

Suma de números pares enteros del 2 al 100:

```
// Sum. Java, ultimo ejemplo ciclo for
// Repeticibn controlada por contador con estructura for
import java.awt.Graphics;
import java.applet.Applet;

public class Sum extends Applet {
public void paint( Graphics g )
{
int sum = 0;
for ( int number = 2; number <= 100; number += 2 )
{
sum += number;
}
g.drawString( "La suma es + sum, 25, 25 );
}
}
```