

Java: Secuencia - Selección - Iteración

Dr. Mario Marcelo Berón

Índice

1 Secuencia

2 Selección

- Sentencia if
- Sentencia: if...else
- Sentencia switch

3 Iteraciones

- Sentencia for
- Sentencia while
- Sentencia do-while
- Sentencias de Ruptura
 - Sentencia break
 - Sentencia continue

Secuencia

Sintaxis

```
<sentencia -1>  
....  
<sentencia -n>
```

Sentencia Compuesta

En una secuencia pueden haber sentencias compuestas (bloques).

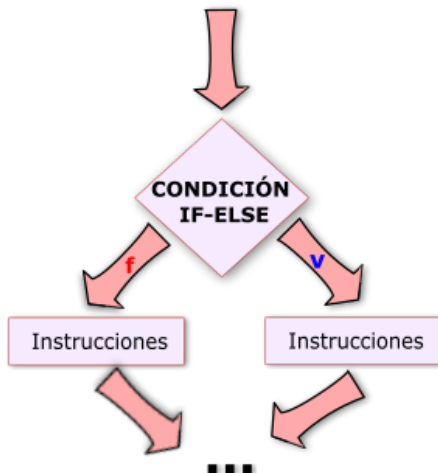
```
{  
    ....  
    <sentencias>  
    ....  
}
```

Secuencia

Sintaxis

```
{  
    System.out.print("La respuesta es: ");  
    System.out.println(r);  
}  
  
{ // Este bloque cambia los valores de x e y  
    int temp; // Una variable temporaria que se usa  
    //en este bloque.  
    temp = x; //Graba una copia del valor de x  
    //en temp.  
    x = y;    //Copia el valor de y en x.  
    y = temp; //Copia el valor de temp en y.  
}
```

Selección

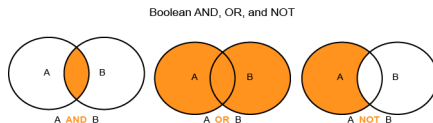


La selección es un método de control de programas en el cual se puede tomar una elección respecto de qué instrucciones ejecutar.

Sentencia if

Operadores Relacionales	
Operador	Significado
==	Igual a
!=	Distinto
<	Menor que
>	Mayor que
≤	Menor o igual que
≥	Mayor o igual que

Operadores Lógicos



Los *Operadores Lógicos* se utilizan para combinar dos o más tests para producir otros tests más complicados.

Operadores Lógicos	
Operador	Significado
AND	&&
OR	
NOT	!

Sentencia if

```
if (cond.)  
{  
    sentencias  
}
```

Si la *cond.* es verdadera (*True*) se ejecutan las sentencias del cuerpo del *if* es decir aquellas que están encerradas entre llaves. En otro caso, el flujo de ejecución continúa sin ejecutarlas. Las expresiones que retornan como resultado *True* o *False* se denominan expresiones *booleanas*.

Atención

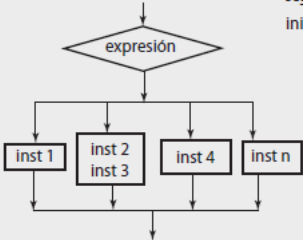
Cuando el cuerpo del *if* tiene una sola sentencia las llaves se pueden omitir.

Sentencia if

```
if (cond.)  
{  
    sentencias  
}  
else  
{  
    sentencias  
}
```

Si la *cond.* es verdadera (True) se ejecutan las sentencias del cuerpo del *if* es decir aquellas que están encerradas entre llaves. En otro caso, se ejecutan las sentencias agrupadas en la parte del *else*.

Sentencia switch

Diagrama de flujo	Pseudocódigo	Lenguaje C
	<pre>segun_sea (expresión) inicio caso exp_const_1: inst1 salir caso exp_const_2: inst2 inst3 salir - - caso exp_const_n: inst n salir caso contrario: inst 10 fin</pre>	<pre>switch (expresión) { case exp_const_1 : inst1; break; case exp_const_2 : inst2 inst3 break; - - case exp_const_n : inst n break; default: inst 10 }</pre>

Una sentencia *switch* se utiliza cuando:

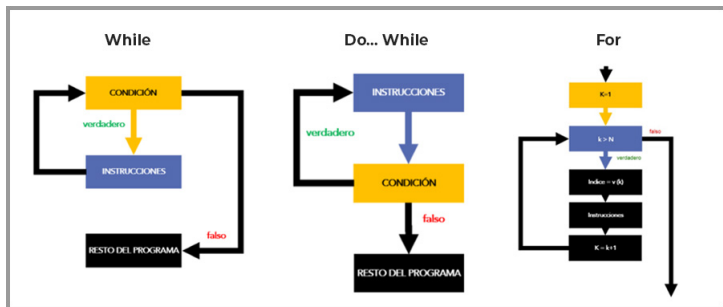
- Solo una variable se verifica en cada condición.
- El chequeo se realiza por valores específicos no por rangos.

Sentencia switch

```
switch( variable )
{
    case valor1 : .....
                break;
    case valor2 : .....
                break;
    default : .....
}
```

- *variable* es el nombre de la variable que se testeará.
- *valor1, valor2,...* son posibles valores de la variable.
- *default* es opcional y se ejecuta siempre que ningún *case* se ejecute.
- El *break* una vez que hubo concordancia salta los *case* restantes.

Iteraciones



Iteración: forma de control de programa que permite instruir a la computadora para que lleve adelante tareas por medio de la repetición de un segmento de código.

- for
- while
- do...while

Iteraciones: for

Se utiliza cuando se desea repetir un segmento de código un número fijo de veces.

```
for (expr1; cond.; expr2)
{
    sentencias
}
```

donde:

- *expr1* es una expresión que por lo general contendrá una inicialización de variables que se utilizan en la condición.
- *cond.* cuando la condición retorna como resultado *True* se ejecutan las sentencias de loop.
- *expr2* es una expresión que se utiliza para modificar las variables que son chequeadas en la condición con el propósito de finalizar el loop.

Iteraciones: while

Se utiliza cuando se desea repetir un segmento de código cero o más veces.

```
while (cond.)  
{  
    sentencias  
}
```

donde:

- *cond.* cuando la condición retorna como resultado *True* se ejecutan las sentencias de loop.
- *sentencias* es el cuerpo del loop *while*. Es el conjunto de sentencias que se desean repetir.

Iteraciones: do-while

Se utiliza cuando se desea repetir un segmento de código una o más veces.

```
do {  
    Sentencias  
}while (cond.);
```

Las sentencias se ejecutan mientras la condición *cond.* retorne como resultado *True*.

Iteraciones: break

Se utiliza para salir de un loop sin que este alcance su fin natural. Se puede utilizar en cualquier tipo de loop.

```
for(expr1; cond1; expr2){  
    .....  
    if (cond2)  
        break;  
}  
sentencia
```

Asuma que el loop se comienza a ejecutar y en una repetición *cond2* retorna como resultado true entonces se ejecuta la sentencia *break*. Esto produce que la próxima sentencia a ejecutar sea *sentencia* en lugar de las sentencias que se encuentran en el cuerpo del loop.

Iteraciones: continue

Se utiliza para forzar al loop a saltar sentencias y continuar con la próxima iteración.

```
for(expr1; cond1; expr2){  
    .....  
    if (cond2)  
        continue;  
}  
sentencia
```

Asuma que el loop se comienza a ejecutar y en una repetición *cond2* retorna como resultado true entonces se vuelve a evaluar la condición. Las sentencias después del *continue* no se ejecutan.