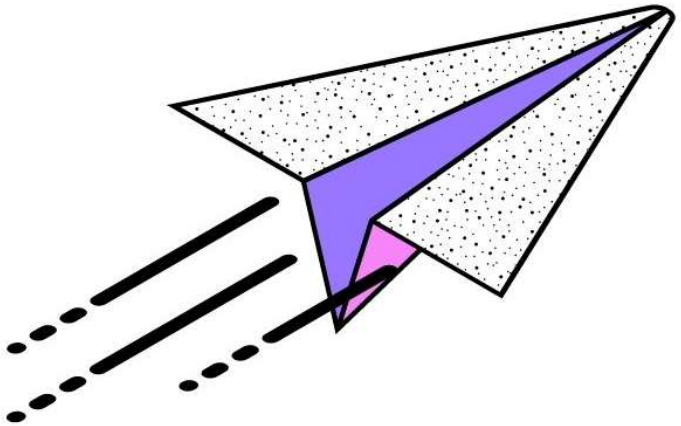
Java 7 SE Fundamentals

Sección 5

Operadores y construcción de sentencias lógicas



- Creación de estructuras if, if/else.
- Sentencias condicionales anidadas y encadenadas.
- Igualdad entre cadenas (Strings).
- Uso de operadores relacionales y condicionales.
- Sentencia **switch.**
- Evaluación de diferentes condiciones en un programa para determinar un algoritmo.

Sentencia if

Una sentencia **if** ejecuta las instrucciones si la condición es verdadera.

Que sentencias que permite Java

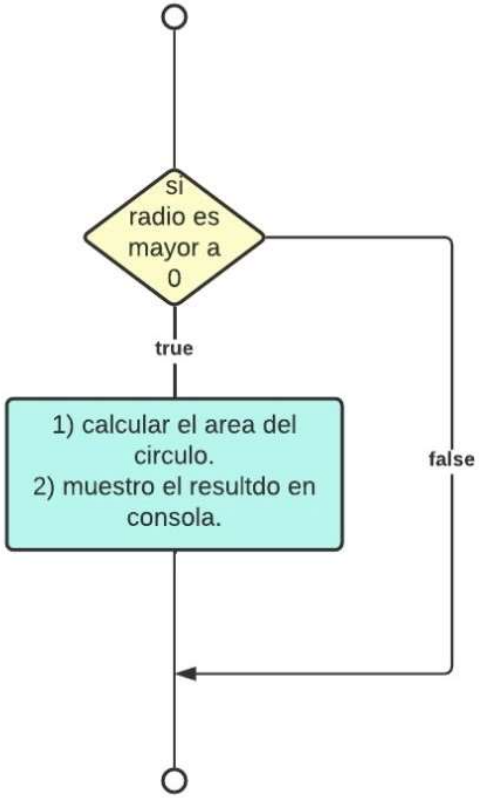
Java tiene varios tipos de sentencias de selección: sentencias if simples, sentencias if-else, sentencias if anidadas, sentencias switch y expresiones condicionales.

Como luce una sentencia if

```
if (expresión-booleana) {
      instrucción(es);
}
```

Diagrama de flujo

como se comporta una sentencia if



Sentencia if-else

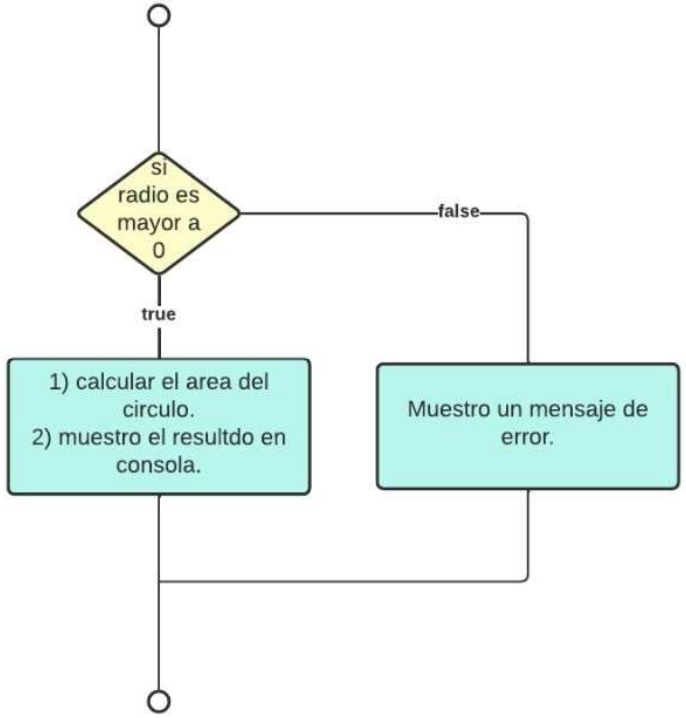
Una sentencia **if** ejecuta las instrucciones delimitadas por ella si la condición es verdadera, en caso contrario ejecutara las contenidas en el bloque delimitado por la palabra **else.**

Como luce una sentencia if-else

```
if (expresión-booleana) {
      instrucción(es); //para el caso true
}else{
      instrucción(es); //para el caso false
}
```

Diagrama de flujo

como se comporta una sentencia if-else



Como luce una sentencia if anidada.

```
if (expresión-booleana 1) {
      if(expresión-booleana 2){
            instrucción(es); // para el caso en que ambas condiciones
// resulten verdaderas
      }
}
```

Como luce una sentencia if encadenada.

```
if (expresión-booleana 1) {

      instrucción(es); // para el caso verdadero

}else  if(expresión-booleana 2){

      instrucción(es); // para el caso falso en 1 y verdadero en 2.
}else{

      instrucción(es); // para el caso falso tanto en 1 como en 2.

}
```

Comparación de cadenas con el operador ==

- Se utiliza para comparar cadenas y objetos en general.
- **==** es un **operador**
- **==** compara la referencia o dirección de memoria del objeto en el Heap, verifica que la dirección sea la misma o no.

Comparacion de cadenas con el metodo .equals()

- Se utiliza también para comparar cadenas y objetos en general.
- .equals() es un método.
- Este no verifica direccion de memoria, este se enfoca en el contenido, en este caso en las letras que componen la cadena.

Operadores relacionales.

Los operadores relacionales son de utilidad para realizar comparaciones y asi determinar si se ejcuta o no cierta sentencia.

Operador	Nombre	Ejemplo var = 5	Resultado
<	Menor que	var < 0	false
<=	Menor o igual que	var <= 0	false
>	Mayor que	var > 0	true
>=	Mayor o igual que	var >= 0	true
==	Igual a	var == 0	false
!=	Distinto de	var != 0	true

Operadores lógicos.

Los operadores lógicos son de utilidad para realizar para evaluar expresiones booleana compuestas de mas de una expresión.

Operador	Nombre	Descripción
!	not	negación lógica
&&	and	conjunción lógica
||	or	disyunción lógica
^	exclusive or	exclusión lógica

Ejercicios practicos

- Desarrollo de un programa que evalue la igualdad entre dos cadenas, utilizando == y equals(), si son iguales imprimir un mensaje en consola indicándo la igualdad, si no, imprimir un mensaje indicando que no lo son.

- Desarrollar un programa que verifique si un numero es par o no.

- Desarrollar un programa que adivine el dia en que nacio una persona.

Sentencia switch

Una sentencia **switch** ejecuta instrucciones basadas en el valor de una variable.

Como luce una sentencia if

```
switch (expresión-switch){

    case valor1: instrucción(es)1;
            break;

    case valor2: instrucción(es)2;
            break;

    case valorN: instrucción(es)N;
            break;

     default: instrucción(es); // ningún caso se cumplió
}
```