

If

Instrucciones condicionales en Java

Cátedra 3

Otoño 2005

La sentencia **if** permite ejecutar una instrucción (o bloque de instrucciones) dependiendo del valor de verdad de una condición, es decir, permite la ejecución condicional. Por ejemplo, **if** permite chequear condiciones como el valor del discriminante de una ecuación cuadrática.

1 Máximo de dos numeros con if

En el siguiente programa se muestra el uso de **if** para calcular el máximo de dos números. No se muestra el método **main**, ni la declaración de la consola.

```
1 con.print ("Ingresa dos números");
2 int x=con.readInt();
3 int y=con.readInt();
4 int máximo=y;
5 if (x>y)
6     máximo=x;
7 con.print("El máximo es "+máximo);
```

1.1 Detalle

4 Se declara la variable máximo que es inicializa con el valor de y

5-6 Si x es mayor que y entonces el valor de x se le asigna a máximo

2 Estructura de if simple

La instrucción **if** más simple consiste en la palabra **if** seguida de una condición entre paréntesis y la instrucción que se debe ejecutar si la condición es verdadera:

```
if (condición)
    instrucción;
```

Si la condición no se cumple, entonces la instrucción no se ejecuta y se pasa a la siguiente.

3 Condiciones

Una condición es una expresión que toma un valor verdadero o falso. Por mientras analizaremos solo las que comparaciones aritméticas entre sí.

Operador	Significado
(a==5)	es a igual a 5?
(a>5)	es a mayor que 5?
(a<5)	es a menor que 5?
(5>=b)	es 5 mayor o igual que b?
(5<=b-a)	es 5 menor que la diferencia entre b y a?
(b!=a)	son distintos a y b?

Las condiciones compuestas se forman a partir de condiciones simples unidas mediante los operadores lógicos **o**, **y** y negación:

Operador	Significado
(a<5 a>8)	es a menor que 5 o mayor que 8?
(nota >5 && nota<7)	es la nota mayor que 5 y menor que 7?
(!(a<3))	es cierto que a no es menor que 3

4 If ... else ...

Muchas veces, lo que nos interesa es ejecutar una instrucción si se cumple una determinada condición y ejecutar otra en caso contrario.

```
1 con.println ("Ingresa el año:");
2 int año=con.readInt();
3 if (año%4==0 && año%100!=0)
4     con.print ("El año es bisiesto");
5 else
6     con.print ("El año es normal");
```

Si seguimos el código

3 se evalúa la condición “¿es el año divisible por 4 y no por 100?”. En realidad es más complicado saber si un año es bisiesto

4 Si la condición es verdadera se escribe “El año es bisiesto” en la pantalla

5-6 Si la condición es falsa, se muestra el mensaje “El año es normal”

5 Bloques de instrucciones

Si uno quiere que se ejecute más de una instrucción si la condición del **if** es verdadera, puede agrupar las instrucciones en un bloque:

```
1 con.print ("Ingresa los coeficientes a, b y c");
2 double a=con.readDouble();
3 double b=con.readDouble();
4 double c=con.readDouble();
5 double discriminante=b*b-4*a*c;
6 if (discriminante>0) {
7     con.print ("Las soluciones son reales y valen ");
8     double x1=(-b+Math.sqrt(discriminante))/(2*a);
9     double x2=(-b-Math.sqrt(discriminante))/(2*a);
10    con.println (x1+" y "+x2);
11 } else if (discriminante==0){
12     con.print("Existe una sola solución real que vale ");
13     double x=-b/(2*a);
14     con.println (x);
15 } else {
16     con.println ("Las soluciones son complejas y valen");
17     double imaginario=Math.sqrt(-discriminante)/(2*a);
18     double real=-b/(2*a);
19     con.print (real+"+i imaginario+i y ");
20     con.print (real+"-i imaginario+i");
21 }
```

Un bloque de instrucciones es un conjunto de instrucciones que actúan como si fueran una sola. Por lo general, Java permite colocar un bloque de instrucciones en el contexto donde se puede colocar una sola instrucción. Un bloque comienza con una llave “{” y termina con “}”.

5.1 Descripción

1-5 Se leen desde el teclado los coeficientes de la ecuación $ax+bx+c=0$ y se calcula el discriminante

7-11 Si el determinante es mayor que 0, entonces se muestran las dos soluciones reales

12-15 Si el determinante es igual a 0, entonces se muestra la única solución real. En la práctica no es buena idea comparar un número de tipo `double` con otro `double`, por problemas de representación interna.

16-21 Si ninguna de las dos condiciones anteriores se ha cumplido, entonces se escribe el valor de las dos soluciones complejas. Ojo!, el computador nunca trabaja internamente con números complejos. Nos las arreglamos con `doubles`. Solo al momento de escribir en la pantalla representamos el complejo agregando una `i`

AL MARGEN

Problemas propuestos

- Escribe un programa que lea un número desde el teclado y escriba si es par o no. Escribe un programa que lea tres números del teclado y los escriba ordenados de menor a mayor

Cuidado con...

Un error común es agregar una cláusula **else** que no hace nada. Aunque Java lo permite, el código se vuelve más complicado de entender. Por ejemplo, la cláusula `else` del siguiente programa está demás. Esta trata de representar "si la velocidad supera los 50 kilómetros por hora, entonces hay que subir el cambio, si no, el cambio se mantiene"

```
if (velocidad>50)
    cambio=cambio+1;
else
    cambio=cambio;
```

Lo correcto es no incluir el `else` :

```
if (velocidad>50)
    cambio=cambio+1;
```