

While

Repetición de instrucciones

Cátedra 4

Otoño 2005

Hasta ahora los programas que hemos visto se caracterizan por el hecho de que las instrucciones se van ejecutando en orden y una sola vez. Sin embargo, Java, al igual que muchos otros lenguajes, permite repetir una cierta cantidad de instrucciones mientras se cumpla una condición. La sentencia se llama a **while** y es similar en su sintaxis a **if**, pero distinta en su semántica.

while se puede usar, por ejemplo, en un programa que lea dos números enteros y escriba todos los números mayores que el primero y menores que el segundo:

```
Ingresar el primer número: 38
Ingresar el segundo: 45
Los números son: 39 40 41 42 43 44
```

1 Flujo de un programa

while es una instrucción de control de flujo cuya sintaxis es:

```
while (condicion) {
    instrucciones;
}
```

while ejecuta instrucciones mientras **condicion** sea verdadera. Si durante la ejecución de las instrucciones la condición se hace falsa, el **while** no termina sino hasta ejecutar la última instrucción del bloque.

Nota En caso que sólo se quiera repetir una sola instrucción, se pueden eliminar los paréntesis:

```
while (condicion)
    instrucción;
```

2 Problemas resueltos

2.1 Intervalo de números

Veamos la solución del problema anterior usando **while**:

```
1 con.print ("Ingresar el primer número: ");
2 int a=con.readInt();
3 con.print ("Ingresar el segundo:");
4 int b=con.readInt();
```

```
5 con.print ("Los numeros son");
6 int i=a+1;
7 while (i<b) {
8     con.print (i+" "); // espacio entre los números
9     i=i+1;
10 }
11 con.println ("Muchas gracias por usar este programa");
```

2.1.1 Siguiendo la ejecución

1-4 Se escriben los mensajes en la pantalla y se leen las respuestas del usuario desde el teclado. Supondremos para el ejemplo que los números son 6 y 9.

7 Como *i* (7) es menor que *b* (9), se ejecutará el bloque de instrucciones.

9 El valor de *i* se incrementa en 1, para quedar en 8

7 Como 8 es menor que 9, se ejecuta nuevamente las líneas 8-9

9 El valor de *i* se incrementa

7 La condición no es verdadera... se pasa a la línea 11

2.2 Una solución alternativa

A veces existen muchas formas de llevar a cabo una tarea. En el caso del programa que acabamos de analizar se puede usar un contador en el ciclo **while**. Un contador es una variable auxiliar que se usa junto a un **while** para repetir un trozo de código una cierta cantidad determinada de veces. En nuestro ejemplo, sabemos que la cantidad de números entre *a* y *b* son *b-a-1*. Por ejemplo, si *a* vale 5 y *b* 8, hay que escribir $8-5-1=2$ números, que son 6 y 7. Tomando esto en cuenta, tenemos una nueva versión del programa:

```
1 int i=b-a-1;
2 while (i>0) {
3     con.print (b-i);
4     i=i-1;
5 }
```

Otra versión, igualmente correcta es:

```
6  int i=b-a-1;
7  while (i>0) {
8      i=i-1;
9      con.print(b-(i+1));
10 }
```

Una ejecución del grupo de instrucciones se denomina iteración (o loop). Si se ejecuta 10 veces el grupo de instrucciones, se dice que el ciclo tuvo 10 iteraciones. No existe un límite al número de iteraciones que se pueden llevar a cabo.

2.3 Sumador

Escribe un programa que lea 10 números desde el teclado y escriba su suma

La solución es

```
int suma=0;
int i=0;
while (i<10) {
    int n=con.readInt();
    suma+=n;
    i++;
}
con.print(suma);
```

3 Ámbito (o “visibilidad”) de una variable

El ámbito de una variable es la parte del programa donde la variable es conocida y puede ser usada (donde la variable “vive”). Como ya vimos, una variable puede ser usada sólo después que se ha definido. En Java, el ámbito de la variable termina cuando se encuentra la llave } que cierra la llave { que precedía su declaración. Cualquier referencia a una variable tiene que ser hecha dentro del ámbito de la variable, si no, Java no compila el programa.

Nota Las variables que son declaradas dentro de un while pierden su valor al final de cada iteración, y se declaran nuevamente en la siguiente (como si se tratara de una nueva variable en cada iteración).

4 Indentación del while

Indentar es ordenar un programa agregando espacios al lado izquierdo de las instrucciones. Esto permite ver fácilmente el flujo del programa.

```
while (condición) {
    instrucción;
    instrucción;
    instrucción;
}
```

Se deja un espacio antes del comienzo de cada instrucción dentro del bloque para clarificar el programa. Nunca olvides indentar. Si bien no es obligación hacerlo, te puede simplificar bastante la mantención del programa.

5 For

En Java existe una forma abreviada de escribir un ciclo while: la instrucción for. No sólo abrevia, sino que además aporta claridad en ciertos casos. Su forma es

```
for (inicialización; condición; repetir) {
    instrucciones;
}
```

y equivale a

```
inicialización;
while (condición) {
    instrucciones;
    repetir;
}
```

AL MARGEN

Problemas propuestos

- Escribe un programa que lea un número desde el teclado y muestre todos sus divisores
- Escribe un programa que para el mismo número muestre todos los factores
- Para el programa 2.3, indica el ámbito de cada variables, es decir, los números de línea donde cada variable es “visible”

Apple e IBM...

- Hay un chiste que dice que IBM creó un computador tan, pero tan rápido, que ejecutaba un loop infinito en 7 segundos.
- Una curiosidad relacionada con **while** es la dirección de Apple Computer en California: Infinite Loop I... está en una rotonda.