

Conociendo Java

Sitio: [Agencia de Aprendizaje a lo largo de la Vida](#)
Curso: Técnicas de Programación - Turno mañana
Libro: Conociendo Java

Imprimido por: Tomas Friz
Día: miércoles, 15 de septiembre de 2021, 13:11

Tabla de contenidos

- 1. Lenguajes y paradigmas
- 2. La codificación
- 3. Los comandos
- 4. Las variables.
- 5. Operadores aritméticos
- 6. Operadores de asignación
- 7. Operadores de igualdad y relacionales
- 8. Operadores lógicos
- 9. Operadores para tipo string
- 10. Operadores de pertenencia

1. Lenguajes y paradigmas



Los lenguajes de programación, entonces, contienen las instrucciones, funciones, librerías y demás recursos para poder generar el código que me permite poner en marcha mi aplicación.

Los lenguajes responden a un **paradigma**.

Pero ¿qué es un paradigma?

Un paradigma es un modelo. Una estructura que contiene reglas y que responde a una convención formada por especialistas, en este caso programadores, que definen el tipo, el método y fijan las normas a seguir. Es un modo determinado estandarizado al cual responden o se adaptan los lenguajes de programación.

Los lenguajes se dividen según varias clasificaciones, las más comunes son:

- Por su Nivel de abstracción:
 - Bajo Nivel: Lenguaje más cercano al código de máquina. Ejemplos: Assembler.
 - Alto Nivel: Lenguaje más cercano al propio lenguaje humano. Ejemplos: Java, Python, Visual Basic.
 - Hay otros lenguajes que pueden tener acceso directo al bajo nivel y manejar instrucciones de alto nivel, como el lenguaje C.
- Por su paradigma:
 - Imperativo: Basado en órdenes, donde cada paso deriva en otro. Los primeros lenguajes utilizaron este paradigma, que en el futuro, se incluye en varios lenguajes modernos.
 - Estructurado: Se divide en bloques, que permiten la reutilización del código generado mediante el uso de procedimientos y funciones.
 - Orientado a Objetos: (hay múltiples definiciones)
 - Funcional: Se basa en expresiones matemáticas.

Existen otras clasificaciones que veremos a lo largo del curso.

2. La codificación

Hasta ahora nos hemos basado en cuestiones abstractas, que nos dan una idea global de cuál es el problema a resolver, cuáles son los requerimientos, los límites y el tiempo que nos llevaría implementar la solución.

Sin embargo, en este punto debemos volver a lo principal. Aplicar los distintos algoritmos de resolución dentro de un programa que de las órdenes necesarias para obtener una solución.

Y la única forma de hacerlo es codificar.

Cómo ya hemos visto, debemos utilizar un [lenguaje de programación](#), para resolver cualquier problema que nos sea dado.

En general, como vimos en capítulos anteriores, tratamos de identificar los elementos que componen al problema, para encontrar una solución, en una serie de pasos lógicos que pueden representarse mediante un diagrama de flujo.

Sin embargo la elección del lenguaje depende del tipo de solución, los requerimientos y el entorno en el que se va a aplicar.

En los lenguajes de programación existen una serie de reglas y estructuras sintácticas que deben respetarse y que son distintas en cada lenguaje.

Los identificadores

También denominados nombres, son una serie de caracteres que pueden ser letras, dígitos, guiones bajos (_) y signos de moneda (\$), que no comience con un dígito ni tenga espacios.

Las variables, las funciones, las clases, los objetos, tienen entonces, un nombre que los identifica. Y ese nombre debe cumplir las reglas antes mencionadas.

Los identificadores no pueden tener el nombre de una palabra clave o reservada, propia del [lenguaje de programación](#).

3. Los comandos



Cada comando es una orden y son palabras reservadas que no pueden usarse como variables, clases, funciones u objetos.

Por ejemplo, la sentencia `for`, que significa “para”, es una palabra reservada. No puedo declarar una variable con esa palabra.

Palabras reservadas en Java:

<code>abstract</code>	<code>assert</code>	<code>boolean</code>	<code>break</code>	<code>byte</code>	<code>case</code>	<code>catch</code>	<code>char</code>
<code>class</code>	<code>const</code>	<code>continue</code>	<code>default</code>	<code>do</code>	<code>double</code>	<code>else</code>	<code>enum</code>
<code>extends</code>	<code>false</code>	<code>final</code>	<code>finally</code>	<code>float</code>	<code>for</code>	<code>goto</code>	<code>if</code>
<code>implements</code>	<code>import</code>	<code>instanceof</code>	<code>int</code>	<code>interface</code>	<code>long</code>	<code>native</code>	<code>new</code>
<code>null</code>	<code>package</code>	<code>private</code>	<code>protected</code>	<code>public</code>	<code>return</code>	<code>short</code>	<code>static</code>
<code>strictfp</code>	<code>String</code>	<code>super</code>	<code>switch</code>	<code>synchronized</code>	<code>this</code>	<code>throw</code>	<code>throws</code>
<code>transient</code>	<code>true</code>	<code>try</code>	<code>void</code>	<code>volatile</code>	<code>while</code>		

4. Las variables.



Ya hemos visto que una variable es un contenedor que tiene una ubicación en memoria que tiene un nombre único y un tipo que la identifica.

Sin embargo, el valor que contienen, puede ser nulo o cero, variar o ser constante.

En Java las variables deben ser inicializadas. Es decir, deben tener un nombre, tipo y guardar un valor (o no).

Además existen tipos primitivos de variables y los tipos por referencia (lo veremos más adelante)

Tipos de Variables:

	Tipo	Descripción
	(JAVA)	
NUMÉRICOS	byte	-128 a +127
	short	-32768 a +32767
	int	-2147483648 a +2147483647
	long	64 bits (-2 ⁶³ a 2 ⁶³ - 1)
	float	32 bits con decimales
	double	64 bits con decimales
ALFANUMÉRICOS	char	Un sólo caracter
	String	Conjunto de caracteres
LÓGICOS	boolean	true o false
ESTRUCTURAS	array	vector
(colecciones de datos)	list	conj. datos duplicados conj. datos no duplicados
	set	
	map	pares de datos (clave y valor)

5. Operadores aritméticos



Los operadores son signos, símbolos o palabras que el intérprete del lenguaje identifica dentro de sus sintaxis para realizar una acción (operación) específica.

JAVA	Descripción
+	suma
-	resta
*	multiplicación
/	división
%	resto

6. Operadores de asignación



Pueden tener operandos numéricos enteros o reales y el tipo específico de resultado numérico dependerá del tipo de éstos

- Asignar un valor
- Realizar la operación algebraica y luego incrementar o decrementar el resultado
- Realizar la operación algebraica y luego acumular el resultado.

JAVA	Descripción
=	asignación
++	incremento
--	decremento
+=	suma acumulada
-=	resta acumulada
*=	producto acumulado
/=	división acumulada
%=	resto acumulado

7. Operadores de igualdad y relacionales



Los operadores de relación evalúan si dos valores/objetos cumplen con una condición específica.

El resultado de esta evaluación es un objeto de tipo booleano.

JAVA	Descripción
==	exactamente igual
!=	distinto de
>	mayor que
<	menor que
>=	mayor o igual que
<=	menor o igual que

8. Operadores lógicos



Estos operadores permiten la realización operaciones lógicas.

Por lo general se realizan con objetos de tipo bool.

JAVA	Descripción
	a or b ¿Se cumplen a o b?
&&	a and b ¿Se cumple a y b?
!	not x Contrario a x

9. Operadores para tipo string



Las variables alfanuméricas son cadenas de caracteres.

Por ejemplo la palabra “hola” equivale a la siguiente expresión:

h+o+l+a

Si juntamos “Juan” + “Perez” obtendremos:

“Juan Perez”

Si queremos repetir un texto, tantas veces, ingresamos:

“hola” * 3

Y obtenemos:

“holaholahola”

JAVA	Descripción
+	suma dos cadenas de caracteres
*	repite una cadena de caracteres

10. Operadores de pertenencia



Los operadores `in` y `not in` evalúan si un objeto se encuentra dentro de otro.

Por ejemplo:

`'a' in 'Hola'`

`'z' not in 'Hola'`

Ambos resultados serán Verdaderos (True).