Variables

1 Definición y tipos de variables

Una variable es un contenedor de información. Imagina una variable como una cajita con una etiqueta que indica su nombre, una cajita en la que se puede introducir un valor. Las variables pueden almacenar valores enteros, números decimales, caracteres, cadenas de caracteres (palabras o frases), etc. El contenido de las variables puede cambiar durante la ejecución del programa, de ahí viene el nombre de "variable".

Java es un lenguaje fuertemente tipado, es decir, es necesario declarar todas las variables que utilizará el programa, indicando siempre el nombre y el tipo de cada una.

El nombre que se le da a una variable es muy importante; intenta usar siempre nombres significativos que, de alguna forma, identifiquen el contenido. Por ejemplo, si usas una variable para almacenar el volumen de una figura, una buena opción sería llamarla volumen; si tienes una variable que almacena la edad de una persona, lo mejor es llamarla edad, y así sucesivamente.

Un programa se lee y se depura mucho mejor si los nombres de las variables se han elegido de forma inteligente.

¿Podrías averiguar qué hace la siguiente línea de código?

```
x = vIp3 * Weerty - zxc;
```

Ahora observa el mismo código pero con otros nombres de variables.

```
precioTotal = cantidad * precio - descuento;
```

Se entiende mucho mejor ¿verdad?

Escribiremos los nombres de variables en formato *lowerCamelCase*. La primera letra se escribe en minúscula y, a continuación, si se utiliza más de una palabra, cada una de ellas empezaría con mayúscula. Por ejemplo, edadMin es un buen nombre para una variable que almacena la edad mínima a la que se puede acceder a un sitio web. Observa que hemos usado una mayúscula para diferenciar dos partes (edad y Min). Puede que en algún libro también encuentres nombres de variables con el carácter de subrayado (_) de tal forma que el nombre de la variable sería edad_min, pero como hemos comentado anteriormente, en este libro nos ceñimos al estándar de Google, que exige el formato *lowerCamelCase*.

No se permiten símbolos como $, %, @, +, -, etc. Puedes usar números en los nombres de variables pero nunca justo al principio; 5x no es un nombre válido pero x5 sí lo es.

No se debe poner una letra mayúscula al comienzo del nombre de una variable para no confundirla con una clase (los nombres de las clases comienzan por mayúscula).

1.1 Enteros (int y long)

Las variables que van a contener números enteros se declaran con int. Veamos un ejemplo.

```
/**
* Uso de variables enteras
*
* @author Nombre del Autor
*/
public class VariablesEnteras {
	public static void main(String[] args) {
		int x; // Se declara la variable x como entera
		x = 5; // Se asigna el valor 5 a la variable x
		System.out.println("El valor actual de mi variable es " + x);
		x = 7; // Se asigna el valor 7 a la variable x
		System.out.println("y ahora es " + x);
	}
}
```

Si pretendemos almacenar valores muy grandes en una variable, usaremos el tipo long en lugar de int.

1.2 Números decimales (double y float)

Usamos los tipos double o float cuando queremos (o esperamos) almacenar números con decimales en las variables.

Ciñéndonos al estándar de Google, no daremos por válidas definiciones de variables como la siguiente (aunque funcionaría sin ningún problema).

```
double x, y;
```

Cada variable se debe definir en una línea diferente. Lo siguiente sí sería correcto.

```
double x;
double y;
```

A continuación tienes un ejemplo completo.

```
/**
* Uso de variables que contienen números decimales
*
* @author Nombre del Autor
*/
public class VariablesConDecimales {
	public static void main(String[] args) {
		double x; // Se declaran las variables x e y
		double y; // de tal forma que puedan almacenar decimales.
		x = 7;
		y = 25.01;
		System.out.println(" x vale " + x);
		System.out.println(" y vale " + y);
	}
}
```

Como puedes ver, también se pueden almacenar números enteros en variables de tipo double.

1.3 Cadenas de caracteres (String)

Las cadenas de caracteres se utilizan para almacenar palabras y frases. Todas las cadenas de caracteres deben ir entrecomilladas.

```
/**
* Uso del tipo String
*
* @author Nombre del Autor
*/
public class UsoDeStrings {
	public static void main(String[] args) {
		String miPalabra = "cerveza";
		String miFrase = "¿dónde está mi cerveza?";
		System.out.println("Una palabra que uso con frecuencia: " + miPalabra);

		System.out.println("Una frase que uso a veces: " + miFrase);
	}
}
```

Como puedes ver, en una cadena de caracteres se pueden almacenar signos de puntuación, espacios y letras con tildes.

2 Operadores artiméticos

En Java se puede operar con las variables de una forma muy parecida a como se hace en matemáticas.

Los operadores aritméticos de Java son los siguientes:

OPERADOR	NOMBRE	EJEMPLO	DESCRIPCIÓN
+	suma	20 + x	suma dos números
-	resta	a - b	resta dos números
*	multiplicación	10 * 7	multiplica dos números
/	división	altura / 2	divide dos números
%	resto (módulo)	5 % 2	resto de la división entera
++	incremento	a++	incrementa en 1 el valor de la variable
- -	decremento	a- -	decrementa en 1 el valor de la variable

A continuación tienes un programa que ilustra el uso de los operadores aritméticos.

```
/**
* Uso de los operadores aritméticos
*
* @author Nombre del Autor
*/
public class UsoDeOperadoresAritmeticos {
	public static void main(String[] args) {
		int x;
		x = 100;
		System.out.println(x + " " + (x + 5) + " " + (x - 5));
		System.out.println((x * 5) + " " + (x / 5) + " " + (x % 5));
	}
}
```

3 Asignación de valores a variables

La sentencia de asignación se utiliza para dar un valor a una variable. En Java (y en la mayoría de lenguajes de programación) se utiliza el símbolo igual (=) para este cometido. Es importante recalcar que una asignación no es una ecuación. Por ejemplo x = 7 + 1 es una asignación en la cual se evalúa la parte derecha 7 + 1, y el resultado de esa evaluación se almacena en la variable que se coloque a la izquierda del igual, es decir, en la x, o lo que es lo mismo, el número 8 se almacena en x. La sentencia x + 1 = 23 * 2 no es una asignación válida ya que en el lado izquierdo debemos tener únicamente un nombre de variable.

Veamos un ejemplo con algunas operaciones y asignaciones.

```
/**
* Operaciones y asignaciones
*
* @author Nombre de Autor
*/
public class Asignaciones {
	public static void main(String[] args) {
		int x = 2;
		int y = 9;
		int sum = x + y;
		System.out.println("La suma de mis variables es " + sum);
		int mul = x * y;
		System.out.println("La multiplicación de mis variables es " + mul);
	}
}
```

2.4 Conversión de tipos (*casting*)

En ocasiones es necesario convertir una variable (o una expresión en general) de un tipo a otro.

Simplemente hay que escribir entre paréntesis el tipo que se quiere obtener. Experimenta con el siguiente programa y observa los diferentes resultados que se obtienen.

```
/**
* Conversión de tipos
*
* @author Nombre Autor
*/
public class ConversionDeTipos {
	public static void main(String[] args) {
		int x = 2;
		int y = 9;
		double division;
		division = (double)y / (double)x;
		//division = y / x; // Comenta esta línea y

		// observa la diferencia.
		System.out.println("El resultado de la división es " + division);
	}
}
```