5. Variables (II).

5.1 Cadenas de caracteres (String)

Las cadenas de caracteres se utilizan para almacenar palabras y frases. Todas las cadenas de caracteres deben ir entrecomilladas mediante el símbolo de comillas dobles (“).

```
//Declaramos las variables, inicianizándolas:
String primeraFrase = "En un lugar de la Mancha,";
String segundaFrase = "de cuyo nombre no quiero acordarme\n";

//Escribimos el valor de las cadenas de caracteres por la salida estándar.
System.out.println(primeraFrase);
System.out.println(segundaFrase);

//Concatenación de dos cadenas.
System.out.println(primeraFrase + segundaFrase);
```

```
En un lugar de la Mancha,
de cuyo nombre no quiero acordarme

En un lugar de la Mancha,de cuyo nombre no quiero acordarme
```

Un cadena de caracteres puede contener cero (cadena vacía) o más caracteres. Nos podríamos encontrar con un código como el siguiente:

```
String cadenaInicial = "";
```

En un principio, parece absurdo disponer de una cadena de caracteres que no contiene ningún carácter, pero más adelante verás la utilidad. Por ejemplo, imagina que estás programando el juego del ahorcado y necesitas tener una cadena donde vayas guardando todas las letras que va probando el jugador; en este caso, cuando empieza el juego, no se ha probado ninguna letra, por tanto, esa cadena estará inicialmente vacía y posteriormente se irá rellenando.

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

Puede que te estés preguntando ¿por qué los tipos int, long, double y float empiezan por minúscula y String empieza por mayúscula? La respuesta es que String es en realidad una clase, no un tipo primitivo. Veremos las clases en profundidad más adelante, en la programación orientada a objetos.

5.2. Caracteres (char)

Un carácter suelto como una letra o un signo de puntuación se puede almacenar en una variable de tipo char. El carácter debe ir entrecomillado utilizando las comillas simples (‘).

Hay que tener en cuenta que no es lo mismo "a" que 'a'. Aunque el contenido en ambos casos es la letra “a”, lo primero es una cadena de caracteres y lo segundo es un carácter. En algunos lenguajes de programación se pueden usar indistintamente las comillas simples y las dobles pero en Java tienen un significado muy distinto.

```java
char letra1 = 'j';
char letra2 = 'a';
char letra3 = 'v';
char letra4 = 'a';
System.out.println("letra1: " + letra1);
System.out.println("letra2: " + letra2);
System.out.println("letra3: " + letra3);
System.out.println("todas las letras juntas: " + letra1 + letra2 + letra3 + letra4);
```

```
letra1: j
letra2: a
letra3: v
todas las letras juntas: java
```

Veamos un comportamiento curioso de char:

```
char letra1 = 'a';
char letra2 = 'b';
System.out.println(letra1);
System.out.println(letra2);
System.out.println(letra1 + letra2);
System.out.println(letra1 + "" + letra2);
```

```
a
b
195
ab
```

Existe una correspondencia entre los tipos char e int de tal forma que la suma de dos caracteres se considera la suma de dos enteros. En el programa anterior, la suma letra1 + letra2 en realidad es la suma de los códigos ASCII de la letra “a” y de la letra “b” que son el 97 y el 98 respectivamente. En Java se podrían escribir cosas como 'a' + 7 sin que diera ningún error. Fíjate que en el ejemplo, para mostrar el contenido de las variables letra1 y letra2 de forma consecutiva se recurre al truco de pintar en medio una cadena de caracteres vacía.

5.3. Resumen de tipos y operadores aritméticos.

TIPO	DESCRIPCIÓN	TAMAÑO	EJEMPLO
boolean	verdadero o falso	1 bit	boolean abierto = true;
byte	número entero	8 bits	byte repeticiones = 22;
char	carácter	16 bits	char letra = ‘a’;
short	número entero	16 bits	short pantalones = 22;
int	número entero	32 bits	int asistentes = 22;
long	número entero	64 bits	long poblacion = 22L;
float	número con decimales	32 bits	float nota = 9.5f;
double	número con decimales	64 bits	double precio = 22.55d;

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

5.4. 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.

```java
int x = 2;
int y = 9;

int divisionEntera;
double divisionFlotante;

divisionEntera = y / x;
divisionFlotante = (double) y / (double) x;

System.out.println("El resultado de la división entera es " + divisionEntera);
System.out.println("El resultado de la división flotante es " + divisionFlotante);
```

```
El resultado de la división entera es 4
El resultado de la división flotante es 4.5
```

Ejercicio 5.1

Escribe un programa en el que se declaren las variables enteras x e y. Asígnales los valores 144 y 999 respectivamente. A continuación, muestra por pantalla el valor de cada variable, la suma, la resta, la división y la multiplicación.

Ejercicio 5.2

Escribe un programa en el que dado un valor en millas almacenado en una variable lo convierta a kilómetros.

Ejercicio 5.3

Escribe un programa en el que dado un valor en pies lo convierta a metros.

Ejercicio 5.4

Escribe un programa que declare 5 variables de tipo char. A continuación, crea otra variable como cadena de caracteres y asígnale como valor la concatenación de las anteriores 5 variables. Por último, muestra la cadena de caracteres por pantalla ¿Qué problemas te encuentras? ¿cómo lo has solucionado?

Ejercicio 5.5

Escribe un programa que declare dos variables con la intensidad y resistencia de una corriente eléctrica. El programa calculará el voltaje.