IES Francisco de Goya

1º DAW. Programación

Unidad de Trabajo 2:Variables, tipos de datos, operadores y comentarios.

Antes de empezar

- Vamos a explicar algunas reglas antes de empezar:

  - **Java diferencia entre MAYÚSCULAS y minúsculas:** es decir, en Java, no es lo mismo “dios” que “DIOS”. Entonces, si llamas a cualquier objeto "dios", si haces referencia a él nuevamente, debes llamarlo "dios", no "Dios" o "DIOS".

  - **Convenio de denominación**: no es obligatorio, pero presta atención a cuándo usar mayúsculas y minúsculas. Por ejemplo, si nombra una "variable", debe comenzar con minúsculas, pero si nombra una clase, debe comenzar con mayúsculas.*No te preocupes por esto, lo aprenderemos durante el curso.*

Sintaxis

- Las variables son como una caja que contiene datos. En esta caja puedes poner un número, un carácter, un texto, etc.
- Por ejemplo:

```
int edad = 28;
```

- En este ejemplo:
  - Estoy creando una variable llamada “edad”, para almacenar la edad de una persona por ejemplo.
  - Estoy diciendo que esta variable va a contener números “enteros” (int)
  - Estoy poniendo dentro de esta variable el valor 28.
  - *Nota: en Java, tenemos que terminar todas las líneas con punto y coma (;).*

Identificadores

- El nombre que le damos a una variable es "**identificador**" (edad).
- Los identificadores deben seguir estas pequeñas reglas:
  - Solo puede contener minúsculas (az), mayúsculas (AZ), números (0-9), guiones bajos (_) o signos de dólar ($).
  - No pueden empezar por un número.
  - Hay algunas palabras que no se pueden usar porque están reservadas para otras cosas en Java, por ejemplo: clase, si...

Desafío2_1

Correcto:	Incorr:
address$nombre	!Addres na,me
apellido1 otherVar	1apellido otherVar#
miVar_	my var

- *No es obligatorio, pero todos los programadores usan minúsculas para los identificadores de variables, comenzando por minúsculas y solo usando mayúsculas si el nombre contiene palabras diferentes, por ejemplo: miSegundoApellido*

Tipos de datos I

- Los tipos de datos definen el tipo de datos almacenados en una variable. En nuestro primer ejemplo era un número entero (int) pero puede ser:

Data Type	Size	Description
byte	1 byte	Stores whole numbers from -128 to 127
short	2 bytes	Stores whole numbers from -32,768 to 32,767
int	4 bytes	Stores whole numbers from -2,147,483,648 to 2,147,483,647
long	8 bytes	Stores whole numbers from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
float	4 bytes	Stores fractional numbers. Sufficient for storing 45 decimal digits
double	8 bytes	Stores fractional numbers. Sufficient for storing 323 decimal digits
boolean	1 bit	Stores true or false values
char	2 bytes	Stores a single character/letter or ASCII values

- Para almacenar un texto de más de un carácter usamos la clase “String” que no es exactamente un “tipo de datos” pero esto no es importante ahora

Tipos de datos II

- Estos son algunos ejemplos de variables con diferentes tipos:

```
byte minuto = 59;
Short kilómetros = 14555;
int playerPrice = 11236241;
long cuatrillon = 4000000000000001;[1]
float peso = 74,9f;[2]
double
verySmall=0.00000000000000000000000000000000000000000000000001;
Boolean notificacionsEnabled = false;
char dniLetra = 'm';[3]
Mensaje de cadena = "¡Hola mundo!";[4]
```

Desafío2_2

Operadores matemáticos

Desafío2_3,2_4

Operador	Descripción	Ejemplo	Resultado
+	Suma	6 + 3	9
-	Resta	6 - 3	3
*	Multiplicación	6*3	18
/	División	6 / 3	2
%	Módulo. Devuelve el resto de una división entera	7 % 3	1[1]
++	Incremento. Suma uno al valor de x.	++x x++	x + 1[2]
--	Decremento. Disminuye el valor de x en uno.	--x x--	x - 1

Operadores relacionales

- Estos son los operadores relacionales:

Operador	Descripción	Ejemplo	Resultado
<	Menor que	3 < 4	Verdadero
>	Mayor que	3 > 4	FALSO
<=	Menor o igual que	3 <= 4	Verdadero
>=	Mayor o igual que	3 >= 4	FALSO
==	Igual a	3 == 4[1]	FALSO
!=	Diferente a (no igual a)	3 != 4	Verdadero

Operadores logicos

- Estos son los operadores lógicos:

Operador	Descripción	Ejemplo	Resultado
&&	Operador Y. Es verdadero si ambos operandos son verdaderos.	true && true True && false False && true False && false	verdadero FALSO FALSO FALSO
||	O operador. Es verdadero si al menos un operando es verdadero.	verdadero || verdadero verdadero || falso falso || verdadero falso || falso	verdadero verdadero verdadero FALSO
!		!verdadero FALSO	FALSO verdadero

Operadores lógicos: ejemplos

- Veamos un ejemplo del uso de operadores lógicos (*No te preocupes en este momento, quedarán más claros en la próxima lección.*):

```
int a = 5;
int b = 6;
// Esto sería falso
if ((a==5) && (b==5)) {
...
// Esto sería verdadero
if ((a==5) || (b==5)) {
...
// Esto sería verdadero
if (!(b==5)) {
```

- ***¡¡¡Presta atención a la diferencia entre = y ==!!!***
  Caerás como moscas☺

tipos de comentarios

- Se utilizan para agregar notas al código (para aclarar) pero no se ejecutan.
- Comentario de una línea:

Desafío2_5

```
// Esta variable mantiene la tasa del IVA
double ivaPerc = 21.0;
```

- Comentario de varias líneas:

```
/* Este es un comentario de varias líneas
 Es útil cuando el comentario es largo. */
```

- Comentario de documentación (este comentario se utiliza para generar documentación del código. Lo veremos en detalle más adelante).[1]

```
/** Esto es para generar documentación.
 @versión 1.1 */
```