CURSO JAVA

CAPITULO 1

CONCEPTOS BÁSICOS

 **Github**  **Linkedin**  **Youtube** 

Palabras Reservadas

Tabla 1.1 Palabras reservadas de Java

abstract	class	final	int	public	this
assert	continue	finally	interface	return	throw
boolean	default	float	long	short	throws
break	do	for	native	static	transient
byte	double	if	new	strictfp	try
case	else	implements	package	super	void
catch	extends	instanceof	protected	synchronized	while

Tipos Primitivos

Tabla 1.2 Tipos primitivos en Java

TIPO	USO	TAMAÑO	RANGO
byte	Entero corto	8 bit	De -128 a127
short	Entero	16 bit	De -32768 a 32767
int	Entero	32 bit	De -2147483648 a 2147483647
long	Entero largo	64 bit	De (+-) 9223372036854775808
float	Real precisión sencilla	32 bit	De -10^{32} a 10^{32}
double	Real precisión doble	64 bit	De -10^{300} a 10^{300}
boolean	lógico	1 bit	true o false
chart	texto	16 bit	Cualquier carácter

Constantes

Las constantes son un tipo de variable especial y que se definen con la palabra **reservada** [final Tipo] y con su **nombre** en **mayúsculas** como **standard** de codificación.

Ejem.. [final double PI = 3.141592;]

[final String NOMBRE_DEFAULT = “No Asignado”;]

Operador de asignación

Tabla 1.3 Operador Asignación

Símbolo	Descripción
=	Asignación

Ejemplos:

```
String nombDefault  = “No Asignado”;
int edadStandard = 18;
```

Operadores Aritméticos

Tabla 1.4 Operadores Aritméticos

Símbolo	Descripción
+	Suma
+	Más unario: positivo
-	Resta
-	Menos unario: negativo
*	Multiplicación
/	División
%	Resto módulo
++	Incremento + 1
--	Decremento - 1

**** Nota:**

Operador unario (-) “Negativo”, convierte un valor positivo en negativo `a = 1; b = -a; b = -1;`

Operador incremento (++), existe pre-incremento y post-incremento `++var || var++`

Operador decremento (--), existe pre-decremento y post-decremento `--var || var--`

Ejemplos:

```
String nombDefault  = “No Asignado”;
int edadStandard = 18;
```

Operadores Relacionales

Tabla 1.5 Operadores Relacionales

Símbolo	Descripción
==	Igual que
!=	Distinto que
<	Menor que
<=	Menor o igual
>	Mayor que
<=	Mayor o igual que

Operadores Lógicos

Tabla 1.6 Operadores Lógicos

Símbolo	Descripción
&&	Operador and: Y
||	Operador or: O
!	Operador not: Negación

Operadores << opera y asigna >>

Tabla 1.7 Operadores opera y asigna

Símbolo	Descripción
+=	Suma y asigna
-=	Resta y asigna
*=	Multiplica y asigna
/=	Divide y asigna
%=	Módulo y asigna

Operador Ternario

Tabla 1.8 Operador Ternario

Símbolo	Descripción
?=	Operador ternario
(3 > 2)?”Mayor”:”Menor”	(expresión lógica)?true:false

Precedencia de Operadores

Tabla 1.9 Precedencia de Operadores

Descripción	Operador
Postfijos	expr++ expr--
Unarios prefijos	++expr --expr +expr -expr !expr
Aritméticos	* / %
Aritméticos	+ -
Relacionales	< <= > >=
Comparación	== !=
AND lógico	&&
OR lógico	||
Ternario	?:
Asignación	= += -= *= / = % = & = ^=

Conversión de tipos

Conversión de tipos **(Cast || Casting)**, lo primero a tener en cuenta es que **ambos** han de **ser** del mismo **tipo,** y dependiendo de la capacidad **pueden perderse** parte de los **datos** en la **conversión**, ejemplo de un tipo **double** a un tipo **int.**

Ejemplo:

```
int x = 2.6;          //    Esto daría un error de asignación de tipos variable int
                            asignándole un valor decimal

int x = (int) 2.6;    //    Se perdería la parte decimal  x = 2
```

Lectura de datos

```
//        Lector/escuchador Object Scanner
          declaramos un nuevo Object Scanner con la palabra reservada “new”
Scanner sc = new Scanner(System.in);

sc.nextInt();          //        Lee un número entero int por teclado
sc.nextDouble();       //        Lee un número real (double)
sc.nextLine(); l       //        Lee una cadena de caracteres hasta que se pulsa intro
```

Salida por Consola

```
//        Lector/escuchador Object Scanner
          declaramos un nuevo Object Scanner con la palabra reservada “new”
Scanner sc = new Scanner(System.in);

//        Importamos la Class Lector/escuchador Object Scanner
import java.util.Scanner;

//        Imprimimos/mostramos literalmente un mensaje [ Cadena | String ]
System.out.print("Cadena de texto a imprimir y/o mostrar");

//        Imprimimos una nueva línea con un mensaje [ Cadena | String ]
System.out.println("Cadena de texto a imprimir y/o mostrar");
```

Programa Principal

```
package curso_Inicio_Capitulo_1;

import java.util.Scanner;
/**
*
* @author nombre del autor
*/
public class Ejemplo_1 {
        /**
        * @param args the command line arguments
        */
        public static void main(String args[]) {     //        Clase Principal Main
                //        Lector/escuchador Object Scanner
                          declaramos un nuevo Object Scanner con la palabra reservada “new”
                Scanner sc = new Scanner(System.in);

                //        Declaramos una variable en el entorno de ejecución del programa
                String mensaje;

                //        Leemos/Asignamos datos introducidos por teclado a través del
                          Object Scanner a la variable mensaje
                mensaje = sc.nextInt();

                //        Imprimimos una nueva línea con el valor[ Cadena | String ]
                          de la variable mensaje
                System.out.println(mensaje);

                //        Imprimimos una nueva línea con longitud de la[ Cadena | String ]
                          mensaje
                System.out.println("La cadena tiene:\t" + mensaje + "caracteres");

                //        Imprimimos una nueva línea con un mensaje [ Cadena | String ]
                System.out.println("Cadena de texto a imprimir y/o mostrar");

                //        Llamamos a la función mostrarMensaje() para imprimir el mensaje
                mostrarMensaje();
        }
        /*  FUNCIONES    */
        public static int calcTotalPatasSpiders(String mensaje){

                //        Devolvemos la longitud de la cadena mensaje con la propiedad lenght
                          accedemos a través de “.”
           return (mensaje.lenght);
        }

        public static void mostrarMensaje(){

           System.out.println("Función que muestra Cadena de texto a imprimir y/o mostrar");

        }
}
```

Github

Linkedin

Youtube

Github **Linkedin** **Youtube**