

Fundamentos de Java

Ejercicios

A continuación, una serie de ejercicios para practicar lo visto en clase:

1) Escribir un programa en java que muestre en pantalla un mensaje con todos los tipos de datos manejados por el lenguaje (de tipo primitivas y de referencia). El mensaje debe lucir algo similar a:

“Esto es un valor entero: (variable entera)”
“Esto es un valor flotante: (variable flotante)”

.
.
.

2) Escribir un programa en java que muestre el valor de una variable primitiva de cualquier tipo y una constante. Mostrar un mensaje para cada una, como en el ejemplo anterior.

3) Escribir un programa en Java que muestre por pantalla el resultado de todas las operaciones aritméticas vistas en clase, usando dos variables de cualquier tipo de dato numérico. Debe mostrarse el resultado de la siguiente forma:

“La suma de (a) + (b) es: (resultado)”
donde (a), (b) y (resultado) son variables declaradas.

.
.
.

4) Escribir un programa en Java que muestre por pantalla el resultado de todas las operaciones de relación vistas en clase, usando dos variables de cualquier tipo de dato. Debe mostrarse el resultado de la siguiente forma:

“Tras comparar (a) (operador de relación) (b), el resultado es: (resultado)”
donde (a), (b) y (resultado) son variables declaradas. (operador de relación es uno de los operadores vistos en clase (<, >, !=, entre otros)).

.
.
.

5) Escribir un programa en java que muestre por pantalla el resultado de todos los tipos de datos tras la realización de un casting explícito. Debe mostrarse el resultado de la siguiente forma:

“El valor de (a), variable tal convertido a tal es: (resultado)”

donde (a) y (resultado) son variables declaradas. "ta" corresponde al tipo de dato original y al que se está convirtiendo.

- 6) Crear un programa que muestre la tabla de multiplicar de un número solicitado por pantalla al usuario.

```
java.util.Scanner  
byte / int / long / short
```

- 7) Realizar el siguiente ejercicio sobre Programación Orientada a Objetos (POO):

1. Crear un clase llamada Persona
2. Crear 2 atributos: nombre y apellido
3. Inicializar las variables en el constructor
4. Crear un método para devolver el nombre y apellido concatenados