

UNIDAD TEMÁTICA 1: Introducción a JAVA

PRACTICOS INDIVIDUALES 1

PARTE 1 (sin computadora)

EJERCICIO 1

Dado el siguiente programa:

```
public static void zoop () {
    baffle ();
    System.out.print ("Vos zacata ");
    baffle ();
}
public static void main (String[] args) {
    System.out.print ("No, yo ");
    zoop ();
    System.out.print ("Yo ");
    baffle ();
}
public static void baffle () {
    System.out.print ("pac");
    ping ();
}
public static void ping () {
    System.out.println (".");
}
```

¿Cuál es la salida? Sé preciso acerca de dónde hay espacios y dónde hay nuevas líneas.

Indicar cuál es la respuesta más correcta: (**\n** denota nueva línea)

EJERCICIO 2

Dado el siguiente código fuente:

```
public class Zumbido {

    public static void desconcertar (String dirigible) {
        System.out.println (dirigible);
        sipo ("ping", -5);
    }

    public static void sipo (String membrillo, int flag) {
        if (flag < 0) {
            System.out.println (membrillo + " sup");
        } else {
            System.out.println ("ik");
            desconcertar (membrillo);
            System.out.println ("muaa-ja-ja-ja");
        }
    }

    public static void main (String[] args) {
        sipo ("traqueteo", 13);
    }
}
```

a) ¿Cuál es la primera sentencia que se **ejecuta**?

- b) Escribir número 2 al lado de la segunda sentencia, un 3 al lado de la que se ejecuta en tercer lugar, y así siguiendo hasta el final del programa. Si una sentencia se ejecuta más de una vez, puede que termine con más de un número al lado.

PARTE 2 (con computadora)

EJERCICIO 3

Muchos cálculos pueden ser expresados de manera concisa usando la operación “*multsuma*”, que toma tres operandos y computa $a*b + c$. Algunos procesadores incluso proveen una implementación de hardware para esta operación para números de punto flotante.

Crear un nuevo programa llamado *Multsuma.java*. Escribir un método llamado *multsuma* que toma tres doubles como parámetros y que imprime el resultado de multisumarlo.

Escribir un método *main* que testee *multsuma* invocándolo con unos pocos parámetros simples, como por ejemplo 1.0, 2.0, 3.0, y después imprima por consola el resultado, que en ese caso debería ser 5.0.

EJERCICIO 4

Ingresa el siguiente código fuente en su proyecto (una clase “Alumno” y varios métodos independientes, que no son de la clase “Alumno”) :

```
public class Alumno {
    private String nombre;
    public Alumno () {
        nombre = null;
    }

    public String getNombreAdmiracion() {
        return nombre.concat("!");
    }

    public static void main (String[] args) {
        Alumno alumno = new Alumno();
        System.out.println(alumno.getNombreAdmiracion());
    }
}

public static int recorrer (String cadena) {
    int res = 0;
    for (int i = 1; i <= cadena.length(); i++) {
        if (cadena.charAt(i) != ' ') {
            res++;
        }
    }
    return res;
}

public static int getValor() {
    int vector[] = { 6, 16, 26,36,46,56,66,76 };
    int idx = 8;
    return vector[idx];
}

public static char getPrimerCaracter(String palabra) {
    String string[] = new String[5];
    return (string[1].charAt(1));
}

public static String paraAString(int a) {
    Object x1 = new Integer(a);
    return (String) (x1) ;
}
```

- a) Indicar el error al ejecutar la clase Alumno y corregirlo. ¿cómo lo detectaste?
- b) Indicar el error al ejecutar el método “recorrer” y corregirlo. ¿cómo lo detectaste?
- c) Indicar el error al ejecutar el método “getValor” y corregirlo. ¿cómo lo detectaste?
- d) Indicar el error al ejecutar el método “getPrimerCaracter” y corregirlo. ¿cómo lo detectaste?
- e) Indicar el error al ejecutar el método “paraAString” y corregirlo. ¿cómo lo detectaste?

EJERCICIO 5 Bucles

Escriba una clase **Contador** y utilice un bucle *while* para mostrar el valor de una variable *contador* que se incrementa de a uno.

Siga los siguientes pasos para crear su clase:

1. Cree una clase llamada *contador* con tres atributos llamados: *MAX_CONT*, *incremento* y *contador*. Asigne el valor 50 a *MAX_CONT* y el valor 1 a *contador* e *incremento*. Asegúrese de *declarar* *MAX_CONT* como una variable "final".
2. Cree un método público *mostrarContador* en la clase, que no reciba parámetros y retorne *void*. Por ejemplo:

```
public void displayCount()
```
3. Cree un bucle *while* en el método con las siguientes características:
 - a. Expresión booleana: Repita si el valor de contador es menor o igual que el valor de *MAX_CONT*.
 - b. Bloque de código:
 - i. Imprima el valor de la variable contador.
 - ii. Incremente el valor de la variable contador con el valor de incremento.
Por ejemplo: `contador = contador + incremento;`
4. ejecute el archivo Contador.java y observe los valores emitidos.
5. Reescriba el algoritmo de manera de usar una sentencia *do-while*. Verifique que hace lo mismo.
6. Reescriba el algoritmo de manera de usar una sentencia *for*. Verifique que hace lo mismo.