

Tarea 5 – Paso de parámetros por valor - referencia

Enunciado

1. Sin usar el IntelliJ, intenta contestar a la siguiente pregunta. ¿Cuál es la salida del siguiente programa? ¿Por qué?

```
class Simple {
    public void print(int x) {
        System.out.println("Valor del parámetro: " + x);
    }
}

public class Alumno {
    public static void main(String[] args) {
        int var = 7;
        Simple simple = new Simple();

        System.out.println("Primer valor de la variable local: " + var);
        simple.print(var);
        System.out.println("Siguiendo valor de la variable local: " + var);
    }
}
```

2. Si cambiamos ligeramente el programa anterior. De forma que, el método invocado print () realiza un cambio en su copia del valor en su parámetro formal x.

```
public void print(int x) {
    System.out.println("Primer valor del parámetro formal: " + x);
    x = 100;
    System.out.println("Siguiendo valor del parámetro formal: " + x );
}
```

¿Cuál es ahora la salida del programa? ¿Por qué se sigue imprimiendo 7 en el main, si hemos modificado el valor de la variable x = 100?

3. Si volvemos a modificar el código anterior por el siguiente código.

```
class Simple {
    public int devolver(int x) {
        System.out.println("Primer valor del parámetro formal: " + x);
        x = 100;
        System.out.println("Siguiendo valor del parámetro formal: " + x );
        return x;
    }
}

public class Alumno {
    public static void main(String[] args) {
        int var = 7;
        int resultado = 0;
        Simple simple = new Simple();

        System.out.println("Primer valor de la variable resultado: " +
resultado);
        resultado = simple.devolver(var);
        System.out.println("Siguiendo valor de la variable resultado: " +
resultado);
    }
}
```

¿Cuál es ahora la salida del programa? ¿Por qué ahora si hemos obtenido el valor 100 en la variable resultado en el main?

4. Sin usar el IntelliJ, ¿qué imprime el siguiente código?

```
class Punto {
    public int x = 1, y = 2;
}

class PuntoTester {
    public void print(Punto punto) {
        punto.x = 45;
        punto.y = 83;
        System.out.println("x: " + punto.x+ " y: " + punto.y);
    }
}

class Tester {
    public static void main ( String[] args ) {
        Punto pt = new Punto();
        System.out.println("x: " + pt.x+ " y: " + pt.y);

        PuntoTester tester = new PuntoTester();
        tester.print(pt);

        System.out.println("x: " + pt.x+ " y: " + pt.y);
    }
}
```

¿Por qué ahora si se cambia el valor de x e y si sucede dentro del método?

5. Identifica si hay errores de sintaxis en las siguientes declaraciones de variables de instancia. Recuerda que algunas partes de las declaraciones son opcionales.
- public boolean isEven;
 - Private boolean isEven;
 - private boolean isOdd
 - public boolean is Odd;
 - string S;
 - public String boolean;
 - private boolean even = 0;
 - private String s = helloWorld;
6. Encuentra el error en la cabecera de los siguientes métodos:
- public String boolean()
 - private void String ()
 - private void myMethod
 - private myMethod()
 - public static void Main (String argv[])
7. Explica la diferencia entre cada uno de los siguientes pares de conceptos.
- Una variable de instancia y una variable local.
 - Getter y setter.
 - Crear una instancia y variable de instancia.
 - Paso por valor y paso por referencia.
8. Define una clase Java llamada NumberCruncher que tenga una sola variable entera como su única variable de instancia. Luego define métodos que realicen las siguientes operaciones en su número: get, doble, triple, cuadrado y cubo. Establece el valor inicial del número con un constructor parametrizado.
9. Agrega a la clase anterior el método main. Úsalo para crear una instancia de NumberCruncher, con un cierto valor inicial, y luego imprime su valor, su doble, triple, cuadrado y cubo.

UD3. Programación orientada a objetos

10. Escribe una definición de clase Java para un objeto Cubo, que tenga un atributo entero para la longitud de su lado. El objeto debe poder informar sobre su área y volumen. El área de la superficie de un cubo es seis veces el área de cualquier lado. El volumen se calcula con el cubo del lado.
11. Escribe una definición de clase Java para un objeto UsuarioCubo que utilizará el objeto Cubo definido en el ejercicio anterior. Esta clase debe crear tres instancias de Cubo, cada una con un lado diferente, y luego informar sus respectivas áreas de superficie y volúmenes. Se le debe pedir al usuario por pantalla los lados que tendrán las 3 instancias de Cubo.
12. Comprueba que imprime el siguiente código. ¿Por qué el valor de la variable data no cambia?

```
class Operation{
    int data=50;

    void cambiar(int data){
        data = data + 100;
    }

    public static void main(String args[]){
        Operation op=new Operation();

        System.out.println("antes del cambio " + op.data);
        op.cambiar(500);
        System.out.println("después " + op.data);
    }
}
```

13. Corrige el código anterior para que cambie el valor.

Entrega

- Contesta a las preguntas en PDF (incluye el código de los ejercicios que lo requieran).