

Tarea 6 – Static en Java

Enunciado

1. Considere el siguiente método estático `duplicate()` a continuación.

```
public static String duplicate(String s) {  
    String t = s + s;  
    return t;  
}
```

¿Qué imprime el siguiente fragmento de código?

```
String s = "Hello";  
s = duplicate(s);  
String t = "Bye";  
t = duplicate(duplicate(duplicate(t)));  
System.out.println(s + t);
```

2. ¿Qué sucede cuando se compila y ejecuta el siguiente programa?

```
class Contador {  
  
    int contador;  
    static int staticContador;  
  
    public Contador() {  
        contador++;  
        staticContador++;  
    }  
  
    public int getContador() {  
        return contador;  
    }  
  
    public static int getStaticContador() {  
        return staticContador;  
    }  
}  
  
public class TestContador {  
  
    public static void main(String[] args) {  
        Contador c1 = new Contador();  
        Contador c2 = new Contador();  
        Contador c3 = new Contador();  
        Contador c4 = new Contador();  
        Contador c5 = new Contador();  
        Contador c6 = new Contador();  
        System.out.println("contador es: " + c6.getContador());  
        System.out.println("staticContador es: " + c6.getStaticContador());  
    }  
}
```

Se ha usado el sexto objeto (c6) de la clase contador para devolver el contador, ¿qué hubiera pasado si hubiéramos usado el primero objeto?

3. Dado el siguiente código, responde a las preguntas.

```
public class Calculo {  
  
    public static double obtenerSuma(double d1, double d2) {  
        return d1 + d2;  
    }  
}  
  
class CalculoTester {  
  
    public static void main(String[] args) {  
        Calculo cal = new Calculo();  
        Calculo cal2 = new Calculo();  
        Calculo cal3 = new Calculo();  
        System.out.println(cal.obtenerSuma(69, 22));  
        System.out.println(cal2.obtenerSuma(69, 22));  
        System.out.println(cal3.obtenerSuma(69, 22));  
    }  
}
```

- ¿Qué se imprime de resultado al ejecutar el código?
 - ¿Es necesario instanciar la clase Calculo? Razona tu respuesta.
 - ¿Cómo puedes evitar que alguien cree una instancia de la clase Calculo?
 - Si se evita instanciar la clase Calculo, ¿cómo se podría acceder al método obtenerSuma?
 - Corrige el código anterior de manera que no se pueda instanciar y se realice la suma.
4. ¿Qué sucede cuando intenta compilar y ejecutar el siguiente programa? Razona tu respuesta.

```
public class Estatica {  
  
    static int x = 6;  
  
    Estatica() {  
        x++;  
    }  
  
    void metodo() {  
        System.out.print("-x" + x);  
    }  
  
    public static void main(String[] args) {  
        Estatica e1, e2, e3, e4;  
        Estatica e5 = new Estatica();  
        Estatica e6 = new Estatica();  
        Estatica e7 = new Estatica();  
        e7.metodo();  
    }  
}
```

5. ¿Qué sucede cuando intenta compilar y ejecutar el siguiente programa? Razona tu respuesta.

```
public class Estatica {  
  
    static int x;  
    int y;
```

```

Estatica() {
    x += 2;
    y++;
}

static int obtenerCuadrado() {
    return x * x;
}

public static void main(String[] args) {
    Estatica e1 = new Estatica();
    Estatica e2 = new Estatica();
    int cuadrado = e1.obtenerCuadrado();
    System.out.print("x " + cuadrado + "- y" + e2.y);
}
}

```

6. La clase Math de Java tiene un método estático que generará un número aleatorio entre 0 y 0.99999999, es decir, entre 0 y 1, sin incluir 1. Mediante el uso de aritmética simple, podemos generar números aleatorios entre dos valores cualesquiera. Por ejemplo, la siguiente declaración asigna un número entero aleatorio entre 1 y 100 a la variable:

```
numeroSecreto = 1 + (int) (Matematicas.random() * 100);
```

Dada esta afirmación, diseña e implementa una aplicación que jugará a un juego de adivinanzas con el usuario. El ordenador generará un número aleatorio entre 1 y 100 y luego le permitirá al usuario adivinar el número, indicándole si su suposición es demasiado alta o demasiado baja. Ten en cuenta que, para este problema, el usuario deberá ingresar números enteros en el teclado.

En el código no puede haber literales.

7. Crea una clase Temperatura. El propósito de la clase de temperatura es realizar las conversiones de temperatura. Para convertir una temperatura Celsius a Fahrenheit o viceversa, no es necesario almacenar el valor de temperatura. Por lo tanto, la clase Temperatura no necesita variables de instancia. Tendrá dos métodos estáticos públicos: uno para convertir de Fahrenheit a Celsius y otro para convertir de Celsius a Fahrenheit. Recuerde que los métodos estáticos están asociados con la clase en lugar de con sus instancias. Por lo tanto, no necesitamos crear una instancia de un objeto Temperatura para usar estos métodos. En su lugar, podemos invocar los métodos a través de la propia clase. Los métodos utilizarán las fórmulas de conversión estándar:

$$F = \frac{9}{5}C + 32 \quad C = \frac{5}{9}(F - 32)$$

Utiliza tipos de datos double.

Crea una clase Main que contenga el método main y prueba que las conversiones funcionen correctamente.

Entrega

- Contesta a las preguntas en PDF (incluye el código de los ejercicios que lo requieran, pega el código, no lo incluyas con capturas de pantalla).