

Ejercicios de Java en exposiciones

Estudiante: Ariel Alejandro Calderon

Asignatura: POO

Fecha: 20/09/2024

1. Tema: Objetos

- **Ejercicio:** Realizar un ejemplo hacer del ciclo de vida de un objeto en Java.
- **Codigo:**

```
package ejerciciodeexposicion;

public class Main {

    public class cicloObjeto {
        private String nombre;

        // Constructor: Creación del objeto
        public cicloObjeto(String nombre) {
            this.nombre = nombre;
            System.out.println("Objeto creado: " + nombre);
        }

        // Método de uso: Mostrar el nombre del objeto
        public void mostrarNombre() {
            System.out.println("El nombre del objeto es: " + nombre);
        }

        // Método finalize: Demuestra la finalización (destrucción) del objeto
        @Override
        protected void finalize() throws Throwable {
            System.out.println("Objeto recolectado (destruido): " + nombre);
            super.finalize();
        }
    }

    public static void main(String[] args) {
        // Creación de un objeto de la clase CicloObjeto
        cicloObjeto objeto1 = new cicloObjeto("Objeto 1");

        // Uso del objeto: Llamar a un método del objeto
        objeto1.mostrarNombre(); // Imprime: El nombre del objeto es:
Objeto 1

        // Finalización del objeto: Eliminar la referencia
        objeto1 = null; // El objeto ya no tiene referencia y está listo
para la recolección

        // Sugerir al Garbage Collector que recolecte los objetos sin
referencia
    }
}
```

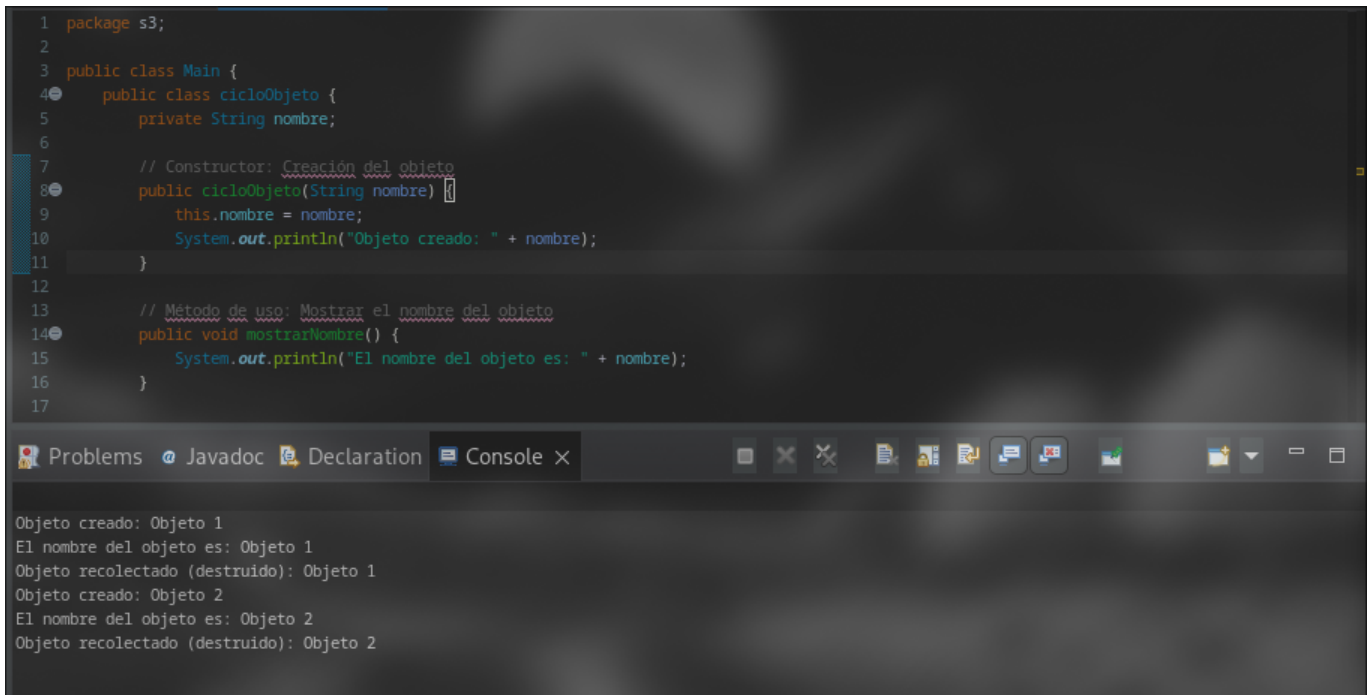
```

        System.gc(); // Sugerencia para activar la recolección de basura

        // Pausa para permitir la acción del Garbage Collector (solo para
observación)
        try {
            Thread.sleep(1000); // Pausa de 1 segundo
        } catch (InterruptedException e) {
            e.printStackTrace();
        }

        // Crear otro objeto para mostrar otro ciclo de vida
        cicloObjeto objeto2 = new cicloObjeto("Objeto 2");
        objeto2.mostrarNombre();
        objeto2 = null; // Dejar el objeto sin referencia
        System.gc(); // Sugerir recolección de basura nuevamente
    }
}

```



```

1 package s3;
2
3 public class Main {
4     public class cicloObjeto {
5         private String nombre;
6
7         // Constructor: Creación del objeto
8         public cicloObjeto(String nombre) {
9             this.nombre = nombre;
10            System.out.println("Objeto creado: " + nombre);
11        }
12
13        // Método de uso: Mostrar el nombre del objeto
14        public void mostrarNombre() {
15            System.out.println("El nombre del objeto es: " + nombre);
16        }
17    }
18 }

```

Objeto creado: Objeto 1
El nombre del objeto es: Objeto 1
Objeto recolectado (destruido): Objeto 1
Objeto creado: Objeto 2
El nombre del objeto es: Objeto 2
Objeto recolectado (destruido): Objeto 2

2. Tema: Atributos

- **Ejercicio:** Crear una clase *Persona* con atributos nombre y edad. Instanciar la clase en dos objetos e imprimir los atributos de la clase.
- **Codigo:**

```

public class Persona {

    String nombre;
    int edad;

    // Método principal

```

```

public static void main(String[] args) {
    // Crear un objeto de la clase Persona
    Persona persona1 = new Persona();
    Persona persona2 = new Persona();

    // Asignar valores a los atributos
    persona1.nombre = "Juan";
    persona1.edad = 18;
    persona2.nombre = "María";
    persona2.edad = 25;

    // Imprimir los valores de los atributos
    System.out.println("Nombre: " + persona1.nombre);
    System.out.println("Edad: " + persona1.edad);
    System.out.println("Nombre: " + persona2.nombre);
    System.out.println("Edad: " + persona2.edad);

}
}

```

The screenshot shows an IDE with a Java file. The code is identical to the one in the first block. The console output at the bottom shows the results of the program execution:

```

<terminated> Person [Java Application] /usr/lib/eclipse/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.linux.x86_64_21.0.4.v20240802-
Nombre: Juan
Edad: 18
Nombre: Maria
Edad: 25

```

3. Tema: Metodos en Java y metodo *main*.

- **Ejercicio:** Implementar dentro de la clase principal un metodo que sume dos enteros y retorne el resultado y otro metodo imprima un mensaje (sin retornar nada). Llamar ambos métodos dentro del metodo main.
- **Codigo:**

```

public class Metodos {

    // Método principal
    public static void main(String[] args) {
        // Llamada a otros métodos
    }
}

```

```

        saludar();
        int resultado = sumar(10, 5);
        System.out.println("El resultado de la suma es: " + resultado);
    }

    // Método sin parámetros y sin retorno
    public static void saludar() {
        System.out.println("¡Hola, bienvenido a mi programa!");
    }

    // Método con parámetros y con retorno
    public static int sumar(int a, int b) {
        return a + b;
    }
}

```

The screenshot shows an IDE with a Java file named 'Metodos.java'. The code defines a class 'Metodos' with three methods: a 'main' method that calls 'saludar' and 'sumar', a 'saludar' method that prints a welcome message, and a 'sumar' method that returns the sum of two integers. The console output shows the results of running the program: '¡Hola, bienvenido a mi programa!' and 'El resultado de la suma es: 15'.

```

2
3 public class Metodos {
4
5     // Método principal
6     public static void main(String[] args) {
7         // llamada a otros métodos
8         saludar();
9         int resultado = sumar(10, 5);
10        System.out.println("El resultado de la suma es: " + resultado);
11    }
12
13    // Método sin parámetros y sin retorno
14    public static void saludar() {
15        System.out.println("¡Hola, bienvenido a mi programa!");
16    }
17
18    // Método con parámetros y con retorno
19    public static int sumar(int a, int b) {
20        return a + b;
21    }
22
23 }

```

Problems Javadoc Declaration Console x

¡Hola, bienvenido a mi programa!
El resultado de la suma es: 15

4. Tema: Parametros y argumentos

- **Ejercicio:** Demuestra cómo un método puede recibir varios parámetros y argumentos. El método "calcularPromedio" calcula el promedio de tres números, recibiendo tres parámetros del tipo double.
- **Código:**

```

public class Parametros {

    public static double calcularPromedio(double num1, double num2, double num3) {
        return (num1 + num2 + num3)/3;
    }

    public static void main(String[] args) {
        double promedio = calcularPromedio(9.0, 8.5, 6.7);
        System.out.println("El promedio calculado es: " + promedio);
    }
}

```

```
}  
}
```

```
1 package s3;  
2  
3 public class Parametros {  
4  
5     public static double calcularPromedio(double num1, double num2, double num3) {  
6         return (num1 + num2 + num3)/3;  
7     }  
8  
9     public static void main(String[] args) {  
10         double promedio = calcularPromedio(9.0, 8.5, 6.7);  
11         System.out.println("El promedio calculado es: " + promedio);  
12     }  
13 }  
14  
15
```

Problems Javadoc Declaration Console x

<terminated> Parametros [Java Application] /usr/lib/eclipse/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.linux.x86_64_21.0.4.v20240
El promedio calculado es: 8.066666666666666

5. Tema: Variables Locales

- **Ejercicio:** Implementar un metodo que reciba tasa de descuento y precio e imprima en pantalla la siguiente informacion: monto de descuento, precio con descuento, monto de impuesto y precio final.
- **Codigo:**

```
public class VariablesLocales {  
    // Variable de instancia (no es local)  
    //Definición: Se declaran dentro de una clase pero fuera de cualquier  
    método.  
    //Alcance: Son accesibles desde cualquier método de la clase en la que  
    se definen.  
    private static final double TASA_IMPUESTO = 0.08;  
  
    public static void main(String[] args) {  
        // Llamada al método calcularPrecioFinal  
        calcularPrecioFinal( 0.15, 500.0);  
    }  
  
    public static void calcularPrecioFinal(double tasaDescuento, double  
precio) {  
        // 'precio' y 'tasaDescuento' son variables locales (parámetros del  
        método)  
  
        // Variable local para almacenar el monto del descuento  
        double montoDescuento = precio * tasaDescuento;  
  
        // Variable local para el precio después del descuento  
        double precioConDescuento = precio - montoDescuento;  
  
        // Variable local para el monto del impuesto
```

```

        double montoImpuesto = precioConDescuento * TASA_IMPUESTO;

        // Variable local para el precio final
        double precioFinal = precioConDescuento + montoImpuesto;

        // Usamos las variables locales para imprimir el resultado
        System.out.println("Precio original: $" + precio);
        System.out.println("Descuento aplicado: $" + montoDescuento);
        System.out.println("Precio con descuento: $" + precioConDescuento);
        System.out.println("Impuesto: $" + montoImpuesto);
        System.out.println("Precio final: $" + precioFinal );

        // Todas las variables locales dejan de existir aquí, al final del
método
    }

}

```

```

1 package s3;
2
3 public class VariablesLocales {
4
5     private static final double TASA_IMPUESTO = 0.08;
6
7     public static void main(String[] args) {
8         // llamada al método calcularPrecioFinal
9         calcularPrecioFinal( 0.15, 500.0);
10    }
11
12    public static void calcularPrecioFinal(double tasaDescuento, double precio) {
13        // 'precio' y 'tasaDescuento' son variables locales (parámetros del método)
14
15        // Variable local para almacenar el monto del descuento
16        double montoDescuento = precio * tasaDescuento;
17
18    }
19 }

```

Problems Javadoc Declaration Console x

<terminated> VariablesLocales [Java Application] /usr/lib/eclipse/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.linux.x86_64_21.0.4.v2

```

Precio original: $500.0
Descuento aplicado: $75.0
Precio con descuento: $425.0
Impuesto: $34.0
Precio final: $459.0

```

6. Tema: Herencia

- **Ejercicio:** Implementa un programa en Java que utilice herencia, donde debes crear una clase llamada Persona que contenga los atributos nombre, apellido y edad, y luego crea una clase que herede de Persona.
- **Código:**

```

public class Herencia {
    public static void main(String[] args) {
        Estudiante estudiante1 = new Estudiante("Alexis", "Guaranda, Via a Chimbo", 23, 85421, 9.8f);
        estudiante1.mostrarDatos();

        // You can create another Estudiante object here if needed
        Estudiante estudiante2 = new Estudiante("Diego", "Guanujo, Via las

```

```

Cochas", 21, 60527, 8.7f);
    estudiante2.mostrarDatos();
}
}

class Persona {
    private String nombre;
    private String direccion;
    private int edad;

    public Persona(String nombre, String direccion, int edad) {
        this.nombre = nombre;
        this.direccion = direccion;
        this.edad = edad;
    }

    public String getNombre() {
        return nombre;
    }

    public String getDireccion() {
        return direccion;
    }

    public int getEdad() {
        return edad;
    }
}

class Estudiante extends Persona {
    private int codigoEstudiante;
    private float notaFinal;

    public Estudiante(String nombre, String direccion, int edad, int
codigoEstudiante, float notaFinal) {
        super(nombre, direccion, edad);
        this.codigoEstudiante = codigoEstudiante;
        this.notaFinal = notaFinal;
    }

    public void mostrarDatos() {
        System.out.println("Nombre: " + getNombre() +
            "\nDireccion: " + getDireccion() + // Fixed the label
            "\nEdad: " + getEdad() +
            "\nCodigo Estudiante: " + codigoEstudiante +
            "\nNota: " + notaFinal + "\n");
    }
}

```

```
40     private int codigoEstudiante;  
41     private float notaFinal;  
42  
43     public Estudiante(String nombre, String direccion, int edad, int codigoEstudiante, float notaFinal) {  
44         super(nombre, direccion, edad);  
45         this.codigoEstudiante = codigoEstudiante;  
46         this.notaFinal = notaFinal;  
47     }  
48  
49     public void mostrarDatos() {  
50         System.out.println("Nombre: " + getNombre() +  
51             "\nDireccion: " + getDireccion() + // Fixed the label  
52             "\nEdad: " + getEdad() +  
53             "\nCodigo Estudiante: " + codigoEstudiante +  
54             "\nNota: " + notaFinal + "\n");  
55     }  
56 }
```

Problems Javadoc Declaration Console ×

```
Nombre: Alexis  
Direccion: Guaranda, Via a Chimbo  
Edad: 23  
Codigo Estudiante: 85421  
Nota: 9.8  
  
Nombre: Diego  
Direccion: Guanujo, Via las Cochas  
Edad: 21  
Codigo Estudiante: 60527  
Nota: 8.7
```