

Métodos y Variables Estáticos

En Java, la palabra clave `static` se utiliza para definir miembros (variables, métodos, bloques y clases anidadas) que pertenecen a la clase en lugar de a una instancia específica de la clase. Esto significa que los miembros `static` son compartidos por todas las instancias de la clase y pueden ser accedidos sin necesidad de crear un objeto de la clase.

Variables estáticas (static variables)

Una variable estática es una variable que pertenece a la clase y no a una instancia específica. Todas las instancias de la clase comparten la misma variable estática.

```
public class Contador {  
    public static int contador = 0;  
  
    public Contador() {  
        contador++;  
    }  
}
```

En este ejemplo, la variable `contador` es compartida por todas las instancias de la clase `Contador`. Cada vez que se crea una nueva instancia, el valor de `contador` se incrementa.

Métodos estáticos (static methods)

Un método estático pertenece a la clase y no a una instancia específica. Puede ser llamado sin necesidad de crear un objeto de la clase.

```
public class Matemáticas {  
    public static int sumar(int a, int b) {  
        return a + b;  
    }  
}
```

Para llamar al método `sumar`, no es necesario crear una instancia de la clase `Matemáticas`:

```
int resultado = Matemáticas.sumar(5, 3);
```

Bloques estáticos (static blocks)

Un bloque estático es un bloque de código que se ejecuta cuando la clase es cargada en la memoria. Se utiliza para inicializar variables estáticas o realizar alguna configuración inicial.

```
public class Ejemplo {
    static {
        System.out.println("Bloque estático ejecutado.");
    }
}
```

Clases anidadas estáticas (static nested classes)

Una clase anidada estática es una clase que está definida dentro de otra clase y es estática. No necesita una instancia de la clase externa para ser creada.

```
public class Externa {
    public static class Anidada {
        public void mostrar() {
            System.out.println("Clase anidada estática.");
        }
    }
}
```

Para crear una instancia de la clase anidada:

```
Externa.Anidada anidada = new Externa.Anidada();
anidada.mostrar();
```

Resumen

- Variables estáticas: Compartidas por todas las instancias de la clase.
- Métodos estáticos: Pueden ser llamados sin crear una instancia de la clase
- Bloques estáticos: Se ejecutan cuando la clase es cargada.
- Clases anidadas estáticas: No necesitan una instancia de la clase externa para ser creadas.

El uso de static es útil cuando quieres que un miembro de la clase sea común a todas las instancias o cuando no necesitas una instancia de la clase para acceder a ese miembro.

Ejercicios

Ejercicio 1: Creación de un Método Estático

Objetivo: Implementar un método estático que calcule el área de un círculo.

```
public class Matematica {
    public static double calcularAreaCirculo(double radio) {
        return Math.PI * radio * radio;
    }
}
```

```
// Uso del método estático
System.out.println("Área del círculo: " + Matematica.calcularAreaCirculo(5));
```

Ejercicio 2: Uso de una Variable Estática

Objetivo: Crear una clase `Empleado` que tenga una variable estática para contar cuántos empleados se han creado.

```
public class Empleado {
    private String nombre;
    public static int contadorEmpleados = 0;

    public Empleado(String nombre) {
        this.nombre = nombre;
        contadorEmpleados++;
    }

    public String getNombre() {
        return nombre;
    }
}

// Creación de instancias y comprobación del contador
Empleado e1 = new Empleado("Ana");
Empleado e2 = new Empleado("Luis");
System.out.println("Número de empleados: " + Empleado.contadorEmpleados); //
Imprime 2
```

Ejercicio 3: Comparación entre Métodos Estáticos y No Estáticos

Objetivo: Implementar métodos estáticos y no estáticos en la misma clase y demostrar sus diferencias.

```
public class Operaciones {
    public static int sumar(int a, int b) {
        return a + b;
    }

    public int multiplicar(int a, int b) {
        return a * b;
    }
}

// Llamada al método estático
int suma = Operaciones.sumar(5, 10);

// Creación de una instancia para llamar al método no estático
Operaciones op = new Operaciones();
int multiplicacion = op.multiplicar(5, 10);
```

Ejercicio 4: Contador de Instancias

Objetivo: Crear una clase `Vehiculo` con una variable estática que cuente cuántos vehículos se han creado.

```
public class Vehiculo {
    private String modelo;
    public static int contadorVehiculos = 0;

    public Vehiculo(String modelo) {
        this.modelo = modelo;
        contadorVehiculos++;
    }

    public String getModelo() {
        return modelo;
    }
}

// Creación de instancias y comprobación del contador
Vehiculo v1 = new Vehiculo("Toyota Corolla");
Vehiculo v2 = new Vehiculo("Honda Civic");
System.out.println("Número de vehículos: " + Vehiculo.contadorVehiculos); //
Imprime 2
```

Ejercicio 5: Uso de Métodos Estáticos para Utilidades

Objetivo: Crear una clase `Convertor` con métodos estáticos para convertir unidades (por ejemplo, de kilómetros a millas).

```
public class Convertor {
    public static double kilometrosAMillas(double kilometros) {
        return kilometros * 0.621371;
    }

    public static double millasAKilometros(double millas) {
        return millas / 0.621371;
    }
}

// Uso de métodos estáticos
System.out.println("10 kilómetros son " + Convertor.kilometrosAMillas(10) + " millas.");
System.out.println("6.2 millas son " + Convertor.millasAKilometros(6.2) + " kilómetros.");
```

Ejercicio 6: Creación de un Banco de Nombres

Objetivo: Crear una clase `NombreBanco` con una variable estática que almacene una lista de nombres y métodos para agregar y mostrar los nombres.

```
import java.util.ArrayList;

public class NombreBanco {
    private static ArrayList<String> nombres = new ArrayList<>();

    public static void agregarNombre(String nombre) {
        nombres.add(nombre);
    }

    public static void mostrarNombres() {
        for (String nombre : nombres) {
            System.out.println(nombre);
        }
    }
}

// Uso de métodos estáticos
NombreBanco.agregarNombre("Carlos");
NombreBanco.agregarNombre("María");
NombreBanco.mostrarNombres();
```

Ejercicio 7: Contador de Palabras

Objetivo: Implementar una clase `ContadorPalabras` con un método estático que cuente el número de palabras en una cadena.

```
public class ContadorPalabras {
    public static int contarPalabras(String frase) {
        if (frase == null || frase.isEmpty()) {
            return 0;
        }
        String[] palabras = frase.split("\\s+");
        return palabras.length;
    }
}

// Uso del método estático
System.out.println("Número de palabras: " + ContadorPalabras.contarPalabras("Hola mundo desde Java"));
```

Ejercicio 8: Validación de Correos Electrónicos

Objetivo: Crear una clase `Validador` con un método estático que valide si una dirección de correo electrónico es válida.

```

public class Validador {
    public static boolean esCorreoValido(String correo) {
        return correo.matches("^[\\w._%+-]+@[\\w.-]+\\.[a-zA-Z]{2,6}$");
    }
}

// Uso del método estático
System.out.println("Correo válido: " +
Validador.esCorreoValido("test@example.com"));
System.out.println("Correo válido: " +
Validador.esCorreoValido("correo@@ejemplo.com"));

```

Ejercicio 9: Generación de Identificadores Únicos

Objetivo: Implementar una clase `IdentificadorUnico` con un método estático que genere un identificador único incremental.

```

public class IdentificadorUnico {
    private static int idActual = 0;

    public static int generarId() {
        return ++idActual;
    }
}

// Uso del método estático
System.out.println("ID generado: " + IdentificadorUnico.generarId());
System.out.println("ID generado: " + IdentificadorUnico.generarId());

```

Ejercicio 10: Calculadora de Factorial

Objetivo: Crear una clase `Calculadora` con un método estático que calcule el factorial de un número.

```

public class Calculadora {
    public static long calcularFactorial(int numero) {
        if (numero <= 1) {
            return 1;
        }
        return numero * calcularFactorial(numero - 1);
    }
}

// Uso del método estático
System.out.println("Factorial de 5: " + Calculadora.calcularFactorial(5));
System.out.println("Factorial de 0: " + Calculadora.calcularFactorial(0));

```

Ejercicio 11: Suma de Números en un Array

Objetivo: Crear una clase `Utilidades` con un método estático que calcule la suma de todos los números en un array.

```
public class Utilidades {
    public static int sumarArray(int[] numeros) {
        int suma = 0;
        for (int numero : numeros) {
            suma += numero;
        }
        return suma;
    }
}

// Uso del método estático
int[] numeros = {1, 2, 3, 4, 5};
System.out.println("Suma de números: " + Utilidades.sumarArray(numeros));
```

Ejercicio 12: Conversión de Temperaturas

Objetivo: Crear una clase `ConversorTemperaturas` con métodos estáticos para convertir entre Celsius y Fahrenheit.

```
public class ConversorTemperaturas {
    public static double celsiusAFahrenheit(double celsius) {
        return (celsius * 9/5) + 32;
    }

    public static double fahrenheitACelsius(double fahrenheit) {
        return (fahrenheit - 32) * 5/9;
    }
}

// Uso de métodos estáticos
System.out.println("25°C a Fahrenheit: " +
    ConversorTemperaturas.celsiusAFahrenheit(25));
System.out.println("77°F a Celsius: " +
    ConversorTemperaturas.fahrenheitACelsius(77));
```

Ejercicio 13: Verificación de Números Primos

Objetivo: Crear una clase `MatematicaAvanzada` con un método estático que determine si un número es primo.

```
public class MatematicaAvanzada {
    public static boolean esPrimo(int numero) {
        if (numero <= 1) {
            return false;
        }
    }
}
```

```

    }
    for (int i = 2; i <= Math.sqrt(numero); i++) {
        if (numero % i == 0) {
            return false;
        }
    }
    return true;
}
}

// Uso del método estático
System.out.println("¿Es primo 7? " + MatematicaAvanzada.esPrimo(7));
System.out.println("¿Es primo 10? " + MatematicaAvanzada.esPrimo(10));

```

Ejercicio 14: Cálculo de Potencias

Objetivo: Implementar una clase `CalculadoraPotencia` con un método estático que calcule la potencia de un número dado.

```

public class CalculadoraPotencia {
    public static double calcularPotencia(double base, int exponente) {
        return Math.pow(base, exponente);
    }
}

// Uso del método estático
System.out.println("2^3: " + CalculadoraPotencia.calcularPotencia(2, 3));
System.out.println("5^4: " + CalculadoraPotencia.calcularPotencia(5, 4));

```

Ejercicio 15: Cálculo de Promedio

Objetivo: Crear una clase `Estadisticas` con un método estático que calcule el promedio de un array de números.

```

public class Estadisticas {
    public static double calcularPromedio(int[] numeros) {
        if (numeros.length == 0) {
            return 0;
        }
        int suma = 0;
        for (int numero : numeros) {
            suma += numero;
        }
        return (double) suma / numeros.length;
    }
}

// Uso del método estático

```



```
int[] datos = {10, 20, 30, 40, 50};  
System.out.println("Promedio: " + Estadisticas.calcularPromedio(datos));
```