

1.Crea una clase llamada Empleado con los atributos nombre, edad y salario. Encapsula los atributos. Crea el método SalarioNeto que calcule el salario con una retención de IRPF del 12,4%.

```
public class Empleado {  
    private String nombre;  
    private int edad;  
    private double salario;  
  
    public Empleado(String nombre, int edad, double salario) {  
        this.nombre = nombre;  
        this.edad = edad;  
        this.salario = salario;  
    }  
  
    public String getNombre() {  
        return nombre;  
    }  
  
    public int getEdad() {  
        return edad;  
    }  
  
    public double getSalario() {  
        return salario;  
    }  
  
    public void setNombre(String nombre) {  
        this.nombre = nombre;  
    }  
  
    public void setEdad(int edad) {  
        this.edad = edad;  
    }  
  
    public void setSalario(double salario) {  
        this.salario = salario;  
    }  
  
    public double calcularSalarioNeto() {  
        double retencion = salario * 0.124;  
        return salario - retencion;  
    }  
}
```

2. Refactoriza el siguiente código. Crea un método ObtenerElMayor que reciba tres números y devuelva el mayor de ellos. (Reducción lógica)

```
public static void main(String[] args) {  
    int a = 5;  
    int b = 7;  
    int c = 2;  
  
    if(a > b && a > c) {  
        System.out.println("a es el mayor");  
    } else if(b > a && b > c) {  
        System.out.println("b es el mayor");  
    } else {  
        System.out.println("c es el mayor");  
    }  
}
```

Refactorizado:

```
public class Entorno {  
  
    public static int obtenerElMayor(int a, int b, int c) {  
        return Math.max(Math.max(a, b), b: c);  
    }  
  
    public static void main(String[] args) {  
        int a = 5;  
        int b = 7;  
        int c = 2;  
  
        int mayor = obtenerElMayor(a, b, c);  
        System.out.println(mayor + " es el mayor");  
    }  
}
```

3. Refactoriza el siguiente código. Crea el método sumar que reciba los números e imprime el resultado del método sumar.

```
public static void main(String[] args) {  
    System.out.println("El resultado de la suma es: " + (2 + 2));  
}
```

Refactorizado

```
public static void sumar(int a, int b) {  
    System.out.println("El resultado de la suma es: " + (a + b));  
}  
  
public static void main(String[] args) {  
    int num1 = 2;  
    int num2 = 2;  
  
    sumar(a: num1, b: num2);  
}
```

4. Refactoriza el siguiente método. Busca la forma de eliminar los bloques duplicados y simplificar la condición.

```
public void checkLogin(String username, String password) {  
    if (username != null && password != null) {  
        if (username.equals(anObject: "admin") && password.equals(anObject: "password")) {  
            // Usuario autenticado  
            System.out.println(x: "Usuario autenticado");  
        } else {  
            // Usuario no autenticado  
            System.out.println(x: "Usuario no autenticado");  
        }  
    } else {  
        // Usuario no autenticado  
        System.out.println(x: "Usuario no autenticado");  
    }  
}
```

Refactorizado

```
public void checkLogin(String username, String password) {  
    if (username != null && password != null && username.equals(anObject: "admin") && password.equals(anObject: "password")) {  
        System.out.println(x: "Usuario autenticado");  
    } else {  
        System.out.println(x: "Usuario no autenticado");  
    }  
}
```

5.Utiliza la expresión condicional ternaria para el siguiente código:

```
public static void main(String[] args) {  
    int x = 5;  
    String result = "";  
  
    if (x > 10) {  
        result = "mayor que 10";  
    } else {  
        result = "menor o igual a 10";  
    }  
  
    System.out.println(x: result);  
  
}
```

Refactorizado

```
public static void main(String[] args) {  
    int x = 5;  
    String result = (x > 10) ? "mayor que 10" : "menor o igual a 10";  
    System.out.println(x: result);  
}
```

6.Refactoriza el siguiente código. Elimina el método sumar para que la operación sea embebida en el método imprimirResultado.

```
int sumar(int a, int b) {  
    return a + b;  
}  
  
void imprimirResultado(int a, int b) {  
    int resultado = sumar(a, b);  
    System.out.println("El resultado de la suma es: " + resultado);  
}  
  
}
```

Refactorizado

```
void imprimirResultado(int a, int b) {  
    int resultado = a + b;  
    System.out.println("El resultado de la suma es: " + resultado);  
}
```

7. Refactoriza el siguiente código. Elimina las variables temporales innecesarias para que el código quede lo más claro posible.

```
public double calcularPrecioTotal(int cantidad, double precioUnitario) {  
    double subtotal = cantidad * precioUnitario;  
    double impuesto = subtotal * 0.16;  
    double total = subtotal + impuesto;  
    return total;  
}
```

Refactorizado

```
public double calcularPrecioTotal(int cantidad, double precioUnitario) {  
    return cantidad * precioUnitario * 1.16;  
}
```