

## Explicación cambios hechos en el código (llenar los \*\*).

### 1. Paquete Compras, clase OrdenCompra:

```
package compras;
import gestionHumana.*;
import java.util.ArrayList;

public class OrdenCompra {
    public int codigo;
    String tipo;
    public Empleado comprador;
    public ArrayList<Producto> productos;

    public OrdenCompra(int codigo, String tipo, Empleado comprador,
        ArrayList<Producto> productos) {
        this.codigo = codigo;
        this.tipo = tipo;
        this.comprador = comprador;
        this.productos = productos;
        Producto.totalProductosPedidos += productos.size();
    }

    public void agregarProducto(Producto producto) {
        if (producto.tipo.equals(tipo)) {
            productos.add(producto);
            Producto.totalProductosPedidos++;
        }
    }

    public void retirarProducto(Empleado empleado, Producto producto) {
        if (!empleado.tengoPermiso()) {
            return;
        }
        retirarProducto(producto);
    }
}
```

```

private void retirarProducto(Producto producto) {
    for (int i = 0; i < productos.size(); i++) {
        if (producto.getCodigo() == productos.get(i).getCodigo()) {
            productos.remove(i);
            producto.totalProductosPedidos--;
            producto.imprimirNombre();
            System.out.println(" retirado");
            break;
        }
    }
}

public void descontar() {
    Producto.totalProductosPedidos -= productos.size();
}
}

```

Entonces,

- Se deben acceder a todas las clases del paquete gestionHumana.
- OrdenCompra es el nombre de esta clase, como se ve en las carpetas.
- Los atributos son de tipo public, default (no se pone nada), public, public. Los public pueden accederse desde cualquier clase y el default solo desde las clases dentro del paquete (OrdenCompra y Producto).
- El método agregarProducto debe ser llamado desde otros paquetes (objtaller3), necesita ser public.
- El método retirarProducto debe ser llamado desde otros paquetes. El return es para seguir de inmediato con la ejecución del código si el if no se cumple.
- Se asegura que i no supere el tamaño de productos y que no haya problemas en el rango, cada índice será válido.
- El método descontar no devuelve nada (void), solo actualiza el contador totalProductosPedidos.

## 2. Paquete Compras, clase Producto:

```
package compras;
```

```
public class Producto {
```

```
    private final int codigo;
```

```

    String nombre;
    public String tipo;
    public static int totalProductosPedidos;

    public Producto(int codigo, String nombre, String tipo) {
        this.codigo = codigo;
        this.nombre = nombre;
        this.tipo = tipo;
    }
    public void imprimirNombre() {
        System.out.print(nombre);
    }
    public void setCodigo(int codigo) {
        // no se escribe nada
    }
    public int getCodigo() {
        return codigo;
    }
    public static int getTotalProductosPedidos() {
        return totalProductosPedidos;
    }
}

```

Entonces,

- Los atributos son de tipo private, default (no se pone nada), public, public. El private solo se puede acceder desde dentro de la clase Producto, el default solo desde las clases dentro del paquete (OrdenCompra y Producto) y los public pueden accederse desde cualquier clase.
- El constructor Producto es public para que otras clases puedan crear productos y trabajar con ellos.
- El método imprimirNombre es public porque otras clases (como OrdenCompra) necesitan invocarlo para imprimir el nombre de un producto. Si este método fuera privado, no se podría llamar desde fuera de la clase Producto.

- No se necesita escribir algo en el método setCodigo. El atributo código está declarado como final, lo que significa que su valor no puede ser modificado después de su inicialización en el constructor.
- codigo es un atributo de tipo int. El método getCodigo recibe a codigo (un int).
- El método getTotalProductosPedidos es public para que cualquier clase que use productos (como OrdenCompra o ObjTaller3) pueda acceder al conteo total de productos pedidos.

### 3. Paquete gestionHumana, clase Empleado:

```
package gestionHumana;

public class Empleado {
    public final long cedula;
    public String nombre;
    String cargo;

    public Empleado(long cedula, String nombre, String cargo) {
        this.cedula = cedula;
        this.nombre = nombre;
        this.cargo = cargo;
    }

    public boolean tengoPermiso() {
        return cargo.contains("Administrador");
    }
}
```

Entonces,

- gestionHumana es el nombre del paquete al que pertenece Empleado, como se ve en las carpetas.
- Los atributos son de tipo public, public, default (no se pone nada). Los public pueden accederse desde cualquier clase y el default solo desde las clases dentro del paquete (Empleado).
- El método tengoPermiso es public porque otras clases necesitan verificar si un empleado tiene permisos para realizar acciones específicas (como retirar productos en OrdenCompra).

#### 4. Paquete objtaller3, clase ObjTaller3:

```
package objtaller3;

import compras.*;
import gestionHumana.Empleado;
import java.util.ArrayList;

public class ObjTaller3 {

    public static void main(String[] args) {
        Producto p1 = new Producto(1, "Escoba", "Aseo");
        Producto p2 = new Producto(2, "Camisa", "Ropa");
        Producto p3 = new Producto(3, "Trapera", "Aseo");
        Producto p4 = new Producto(4, "Pantalon", "Ropa");
        Producto p5 = new Producto(5, "Jabon", "Aseo");
        Empleado emp1 = new Empleado(405, "Juan", "Ingeniero");
        ArrayList<Producto> productos1 = new ArrayList<>();
        productos1.add(p1);
        productos1.add(p3);
        OrdenCompra orden1 = new OrdenCompra(101, "Aseo", emp1, productos1);
        System.out.println(Producto.getTotalProductosPedidos());
        orden1.agregarProducto(p4);
        System.out.println(Producto.getTotalProductosPedidos());
        orden1.agregarProducto(p5);
        System.out.println(Producto.getTotalProductosPedidos());
        System.out.println("Orden " + orden1.codigo + " creada");

        Empleado emp2 = new Empleado(128, "Susana", "Administradora de sucursal");
        ArrayList<Producto> productos2 = new ArrayList<>();
        productos2.add(p2);
        productos2.add(p4);
        OrdenCompra orden2 = new OrdenCompra(202, "Ropa", emp2, productos2);
        System.out.println(Producto.getTotalProductosPedidos());
        System.out.println(emp2.cedula + " va a retirar producto");
    }
}
```

```

orden2.retirarProducto(emp2, p4);
System.out.println(Producto.getTotalProductosPedidos());
orden2.retirarProducto(emp1, p2);
System.out.println(Producto.getTotalProductosPedidos());
}
}

```

Entonces,

- Se deben importar todas las clases dentro del paquete compras (OrdenCompra y Producto).
- Se debe importar, manual y explícitamente, ArrayList (una clase del paquete java.util) por la definicion de productos1 y productos2.
- El método main define sus argumentos de esa forma. Es el punto de entrada estándar para cualquier programa Java.
- La orden de compra tiene código 101, según la salida esperada del programa.
- Se llama al método agregarProducto para actualizar la lista de productos de la orden (productos1) y el contador global (totalProductosPedidos) en la clase Producto.
- El empleado que intenta retirar el producto es representado por emp1. Aunque emp1 no tiene permisos, la salida esperada del programa no se ve afectada: no se incluye ninguna línea que indique un intento fallido de retirar un producto, solo se imprimen los casos donde se realiza una acción exitosa.

Aunque el código corre y ejecuta la salida esperada, hay 2 problemas en la consola de sintaxis que se deben tener en cuenta:

- the static field Producto.totalProductosPedidos should be accessed in a static way: se intenta acceder a un atributo estático de una clase desde un objeto de esa clase. La forma correcta es hacerlo directamente a través del nombre de la clase.
- Accessing static field totalProductosPedidos: es un aviso. El atributo es compartido por todos los objetos de la clase y cambiar su valor puede afectar el comportamiento de otros objetos.