

Fabián Andrés Hurtado Arango

Respuestas Ejercicio 2 - Taller 3 – Java

Ejercicio:

Actividad: Complete los espacios, es ** que se encuentran en el código, de modo que se imprima el siguiente resultado en pantalla y los atributos y métodos estén lo más encapsulados posible (cumpliendo con el principio de encapsulamiento).

Nota: no puede modificar otras secciones, ni agregar o eliminar código.

Salida esperada

```
2
2
3
Orden 101 creada
5
128 va a retirar producto
Pantalon retirado
4
4
```

Paquete compras, clase OrdenCompra.

```
1
2 package compras;
3
4 import gestionHumana.Empleado;
5 import java.util.ArrayList;
6
7 public class OrdenCompra { //Nombre del archivo
8
9     public int codigo; //Se necesita accederlo en otra clase de otro paquete
10    private String tipo; //No se usa fuera de la clase
11    private Empleado comprador; //No se usa fuera de la clase
12    private ArrayList<Producto> productos; //No se usa fuera de la clase
13
14    public OrdenCompra(int codigo, String tipo, Empleado comprador,
15        ArrayList<Producto> productos) {
16        this.codigo = codigo;
17        this.tipo = tipo;
18        this.comprador = comprador;
19        this.productos = productos;
20        Producto.totalProductosPedidos += productos.size();
21    }
22
23    public void agregarProducto(Producto producto) { //Necesita ser accedido desde otra clase de otro paquete
24        if (producto.tipo.equals(tipo)) {
25            productos.add(producto);
26            Producto.totalProductosPedidos++;
27        }
28    }
29
30    public void retirarProducto(Empleado empleado, Producto producto) { //Necesita ser accedido desde otra clase de otro paquete
31        if (!empleado.tengoPermiso()) {
32            return; //Hace que termine el metodo
33        }
34        retirarProducto(producto);
35    }
36
37    private void retirarProducto(Producto producto) { //No se usa fuera de la clase
38        for (int i = 0; i < productos.size(); i++) { //Itera a lo largo de la lista
39            if (producto.getCodigo() == productos.get(i).getCodigo()) {
40                productos.remove(i);
41                producto.totalProductosPedidos--;
42                producto.imprimirNombre();
43                System.out.println(x: " retirado");
44                break;
45            }
46        }
47    }
48
49    public void descontar() { //Retorna null
50        Producto.totalProductosPedidos -= productos.size();
51    }
52
53 }
```

Paquete compras, clase Producto.

```
1 package compras;
2
3 public class Producto {
4
5     private final int codigo; //No se usa fuera de la clase
6     private String nombre; //No se usa fuera de la clase
7     String tipo; //Necesita ser accedido en otra clase del mismo paquete
8     static int totalProductosPedidos; //Atributo de clase
9
10    public Producto(int codigo, String nombre, String tipo) { //Necesita ser accedido en otra clase de otro paquete
11        this.codigo = codigo;
12        this.nombre = nombre;
13        this.tipo = tipo;
14    }
15
16    void imprimirNombre() { //Default para ser conocido en todo el paquete
17        System.out.print(nombre);
18    }
19
20    public void setCodigo(int codigo) {
21        return; //pass
22    }
23
24    public int getCodigo() { //Codigo es de tipo entero
25        return codigo;
26    }
27
28    public static int getTotalProductosPedidos() { //Se necesita accederlo en otra clase de otro paquete
29        return totalProductosPedidos;
30    }
31 }
```

Paquete gestionHumana, clase Empleado.

```
1 package gestionHumana; //Nombre de la 'carpeta'
2
3 public class Empleado {
4
5     public final long cedula; //Necesita ser accedido en otra clase de otro paquete
6     private String nombre; //No necesita ser accedido en otra clase de otro paquete
7     private String cargo; //No necesita ser accedido en otra clase de otro paquete
8
9     public Empleado(long cedula, String nombre, String cargo) {
10        this.cedula = cedula;
11        this.nombre = nombre;
12        this.cargo = cargo;
13    }
14
15    public boolean tengoPermiso() { //Necesita ser accedida desde otra clase de otro paquete
16        return cargo.contains(s:"Administrador");
17    }
18 }
```

Paquete objtaller3, clase ObjTaller3.

```
1 package objtaller3;
2
3 import compras.*; //Importa todas las clases publicas del paquete compras
4 import gestionHumana.Empleado;
5 import java.util.ArrayList; //Permite hacer y manipular Arrays
6
7 public class ObjTaller3 {
8
9     Run | Debug | Run main | Debug main
10    public static void main(String[] args) { //Metodo main
11        Producto p1 = new Producto(codigo:1, nombre:"Escoba", tipo:"Aseo");
12        Producto p2 = new Producto(codigo:2, nombre:"Camisa", tipo:"Ropa");
13        Producto p3 = new Producto(codigo:3, nombre:"Trapera", tipo:"Aseo");
14        Producto p4 = new Producto(codigo:4, nombre:"Pantalon", tipo:"Ropa");
15        Producto p5 = new Producto(codigo:5, nombre:"Jabon", tipo:"Aseo");
16        Empleado emp1 = new Empleado(cedula:405, nombre:"Juan", cargo:"Ingeniero");
17        ArrayList<Producto> productos1 = new ArrayList<>();
18        productos1.add(p1);
19        productos1.add(p3);
20        OrdenCompra orden1 = new OrdenCompra(codigo:101, tipo:"Aseo", emp1, productos1); //Se necesita agregar el codigo de la orden
21        System.out.println(Producto.getTotalProductosPedidos());
22        orden1.agregarProducto(p4);
23        System.out.println(Producto.getTotalProductosPedidos());
24        orden1.agregarProducto(p5); //Se agrega al Array de productos
25        System.out.println(Producto.getTotalProductosPedidos());
26        System.out.println("Orden " + orden1.codigo + " creada");
27
28        Empleado emp2 = new Empleado(cedula:128,nombre:"Susana", cargo:"Administradora de sucursal");
29        ArrayList<Producto> productos2 = new ArrayList<>();
30        productos2.add(p2);
31        productos2.add(p4);
32        OrdenCompra orden2 = new OrdenCompra(codigo:202, tipo:"Ropa", emp2, productos2);
33        System.out.println(Producto.getTotalProductosPedidos());
34        System.out.println(emp2.cedula + " va a retirar producto");
35        orden2.retirarProducto(emp2, p4);
36        System.out.println(Producto.getTotalProductosPedidos());
37        orden2.retirarProducto(emp1, p2); //Se necesita un empleado para retirar producto, pero emp1 no lo puede retirar
38        System.out.println(Producto.getTotalProductosPedidos());
39    }
```

Salida del código:

```
2
2
3
Orden 101 creada
5
128 va a retirar producto
Pantalon retirado
4
4
PS C:\Users\fabi2\OneDrive\Escritorio\ejercicio-2-taller-3-java-fhurtadoa> □
```