

UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE



Object-oriented programming

NRC: 9642

Grupo 1

Chipe Pamela

Cobeña Joan

Davila Anabel

Espin Andres

Topic: Pitfalls Inspection

ING. Jorge Edison Lascano.

Evidence:

DUPLICATED CODE

```
public Dress getDressInformation() {
    System.out.print(s: "Ingrese el nombre del vestido: ");
    String name = scanner.getNoSpacedString();

    System.out.print(s: "Ingrese la marca del vestido: ");
    String brand = scanner.getNoSpacedString();

    System.out.print(s: "Ingrese las medidas del vestido: ");
    String size = scanner.getNoSpacedString();

    System.out.print(s: "Ingrese el precio del vestido: ");
    float price = scanner.getValidFloat();

    System.out.print(s: "Ingrese la cantidad de vestidos: ");
    int quantity = scanner.getValidInt();

    return new Dress(name, brand, size, price, quantity);
}

public Dress getUpdatedDressInformation() {
    System.out.print(s: "Enter New Dress Name: ");
    String name = scanner.getNoSpacedString();

    System.out.print(s: "Enter New Dress Brand: ");
    String brand = scanner.getNoSpacedString();

    System.out.print(s: "Enter New Dress Size: ");
    String size = scanner.getNoSpacedString();

    System.out.print(s: "Enter New Price: ");
    float price = scanner.getValidFloat();
}
```

DEAD CODE

```
private String name;  
private String brand;  
private String size;  
private float price;  
private int quantity;  
private LocalDate dateReceived;  
//private LocalDate dateSold;
```

```
/*public LocalDate getDateSold() {  
    return dateSold;  
}*/  
  
/*public void setDateSold(LocalDate dateSold) {  
    this.dateSold = dateSold;  
}*/
```

MEANINGLESS COMMENTS

```
public int getValidInt() {  
    int number = 0;  
    try {  
        number = scanner.nextInt();  
    } catch (InputMismatchException e) {  
        System.err.println(x: "Ingrese solamente valores enteros");  
        System.out.print(s: "Ingrese el número nuevamente: ");  
        scanner.nextLine(); // Limpiar el búfer del scanner  
        number = getValidInt(); // Llamada recursiva para solicitar el número nuevamente  
    }  
    return number;  
}  
  
public float getValidFloat() {  
    float number = 0;  
    try {  
        number = scanner.nextFloat();  
    } catch (InputMismatchException e) {  
        System.err.println(x: "Ingrese solamente valores numéricos");  
        System.out.print(s: "Ingrese el número nuevamente: ");  
        scanner.nextLine(); // Limpiar el búfer del scanner  
        number = getValidFloat(); // Llamada recursiva para solicitar el número nuevamente  
    }  
    return number;  
}
```

```

public String getNoSpacedString() {
    String text = "";
    try {
        text = scanner.next();
    } catch (InputMismatchException e) {
        System.err.println(x: "Ingrese solamente texto");
        System.out.print(s: "Ingrese el texto nuevamente: ");
        scanner.nextLine(); // Limpiar el búfer del scanner
        text = getNoSpacedString(); // Llamada recursiva para solicitar el texto nuevamente
    }
    return text;
}

```

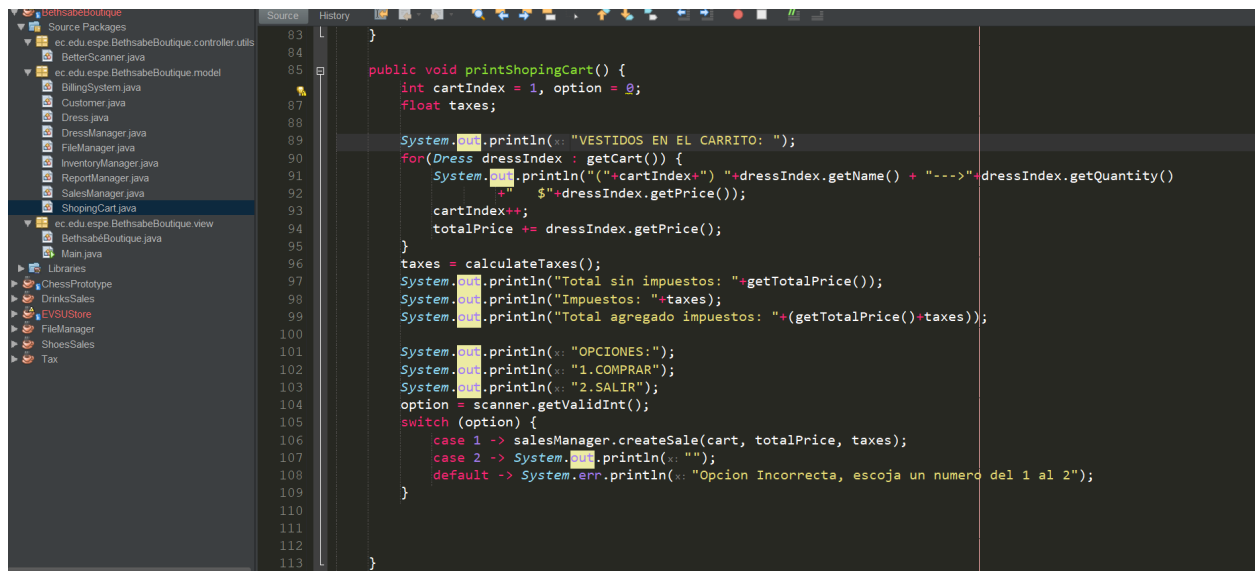
```

public void createCsv(ArrayList<Dress> dressList, String reportName) {
    //Creates the folder "Reportes" in root
    File file = new File(pathname: "Reportes");
    file.mkdir();
    String[] reportsList = file.list(); //Obtains all the reports in "Reportes"

    int dressIndex=0;
}

```

CLASS EXPLOSION



```

83 }
84
85 public void printShoppingCart() {
86     int cartIndex = 1, option = 0;
87     float taxes;
88
89     System.out.println(x: "VESTIDOS EN EL CARRITO: ");
90     for(Dress dressIndex : getCart()) {
91         System.out.println("+"+cartIndex+" "+dressIndex.getName() + "--->"+dressIndex.getQuantity()
92             +" $"+dressIndex.getPrice());
93         cartIndex++;
94         totalPrice += dressIndex.getPrice();
95     }
96     taxes = calculateTaxes();
97     System.out.println("Total sin impuestos: "+getTotalPrice());
98     System.out.println("Impuestos: "+taxes);
99     System.out.println("Total agregado impuestos: "+(getTotalPrice()+taxes));
100
101     System.out.println(x: "OPCIONES:");
102     System.out.println(x: "1.COMPRAR");
103     System.out.println(x: "2.SALIR");
104     option = scanner.getValidInt();
105     switch (option) {
106         case 1 -> salesManager.createSale(cart, totalPrice, taxes);
107         case 2 -> System.out.println(x: "");
108         default -> System.err.println(x: "Opcion Incorrecta, escoja un numero del 1 al 2");
109     }
110
111
112
113 }

```

```

public void removeFromCart() {
    int dressOption;
    int menuOption;
    Dress dress;

    System.out.println("1.Borrar una cantidad");
    System.out.println("2. Borrar todo el carrito");
    menuOption = scanner.getValidInt();

    switch (menuOption){
        case 1:
            printShopingCart();
            System.out.println("Seleccione el numero del vestido a borrar: ");
            dressOption = scanner.getValidInt();
            dress = getCart().get(dressOption-1);

            System.out.println("Ingresa la cantidad a borrar: ");
            int quantity = scanner.getValidInt();
            quantity = dress.getQuantity() - quantity;
            if(quantity == 0){
                cart.remove(0: dress);
            }else{
                dress.setQuantity(quantity);
            }
            break;

        case 2:
            cart.clear();
            break;
    }
}

```

Doing the menus inside a Class in the model