

Engenharia de Software — 2022.1

Equipe:

Data: 16/05/2022
Professora: Diana Braga

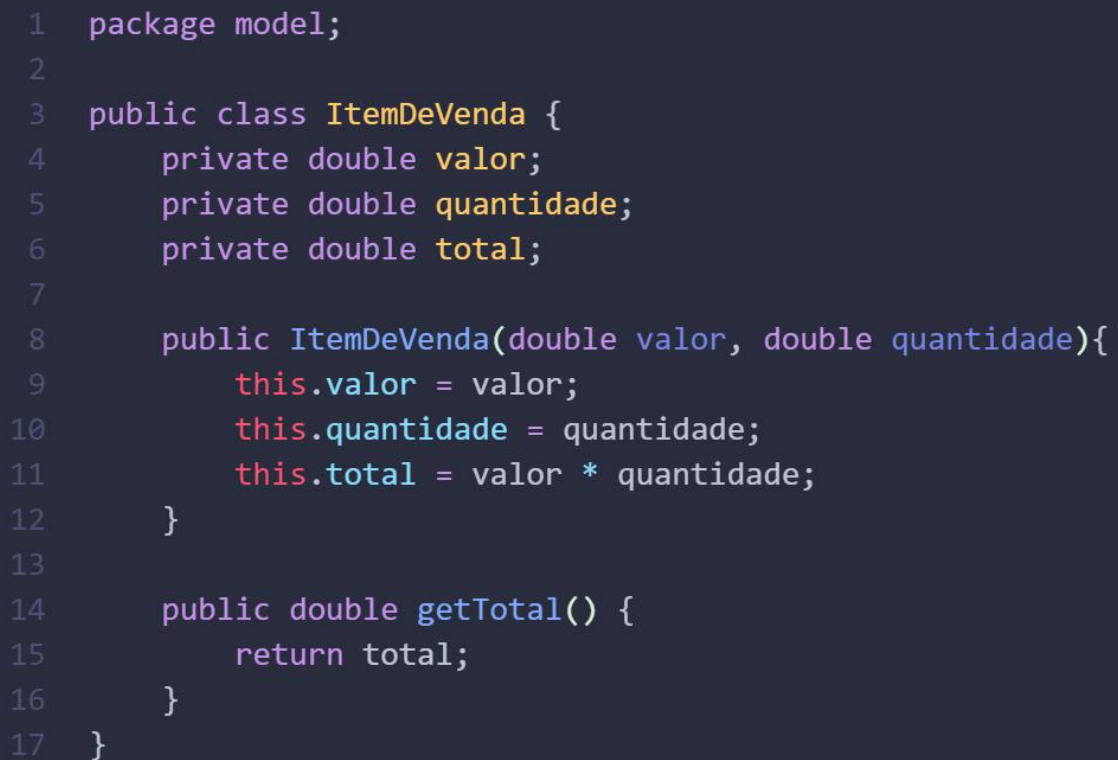
Questão 01 - O princípio de software que não é utilizada é a coesão, haja vista que a mesma especifica que os comportamentos devem ser separados de acordo com os seus objetivos.

Questão 02 – O princípio SOLID usado para melhorar o código a seguir é o Single-responsibility principle – princípio de Responsabilidade Única.

```
1 package controller;
2
3 import java.util.ArrayList;
4 import java.util.List;
5
6 public class Venda {
7     private List<Double> itemValores = new ArrayList<Double>();
8     private List<Double> itemQuantidades = new ArrayList<Double>();
9
10    public double total() {
11        double soma = 0;
12        for(int i = 0; i < itemValores.size(); i++){
13            soma += itemValores.get(i) * itemQuantidades.get(i);
14        }
15        return soma;
16    }
17
18    public void adicionarItem(double valor) {
19        itemValores.add(valor);
20        itemQuantidades.add(1.0);
21    }
22
23    public void adicionarItem(double valor, double quantidade) {
24        itemValores.add(valor);
25        itemQuantidades.add(quantidade);
26    }
27 }
```

No código acima, o princípio de Responsabilidade Única não está sendo aplicado, pois as informações dos itens vendidos estão sendo mantidos na classe Venda.

Para remover essa responsabilidade da classe Venda de manter as informações dos itens criamos uma nova classe chamada ItemDeVenda.



```
1  package model;
2
3  public class ItemDeVenda {
4      private double valor;
5      private double quantidade;
6      private double total;
7
8      public ItemDeVenda(double valor, double quantidade){
9          this.valor = valor;
10         this.quantidade = quantidade;
11         this.total = valor * quantidade;
12     }
13
14     public double getTotal() {
15         return total;
16     }
17 }
```



```
1  package controller;
2
3  import java.util.List;
4  import java.util.ArrayList;
5  import model.ItemDeVenda;
6
7  public class Venda {
8
9      private List<ItemDeVenda> itens = new ArrayList();
10
11     public double total() {
12         double soma = 0;
13         if(itens != null){
14             for (int i = 0; i < this.itens.size(); i++){
15                 soma += itens.get(i).getTotal();
16             }
17         }
18         return soma;
19     }
20
21     public void adicionarItem(double valor) {
22         itens.add(new ItemDeVenda(valor, 1.0));
23     }
24
25     public void adicionarItem(double valor, double quantidade) {
26         itens.add(new ItemDeVenda(valor, quantidade));
27     }
28 }
```