

**CENTRO UNIVERSITÁRIO FACULDADES
INTEGRADAS ESPÍRITO SANTENSES – FAESA
CURSO TECNÓLOGO EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS – EAD**

JEFFERSON SCHITTINI GARCIA

TRABALHO DA DISCIPLINA PROGRAMAÇÃO ORIENTADA A OBJETOS

Vitória, ES

2022

JEFFERSON SCHITTINI GARCIA

TRABALHO DA DISCIPLINA PROGRAMAÇÃO ORIENTADA A OBJETOS

Trabalho acadêmico do Curso Tecnólogo em Análise e Desenvolvimento de Sistemas, apresentado às Faculdades Integradas Espírito Santenses como parte das exigências da disciplina Fundamentos de Redes de Computadores.

Orientadora: professora Lorena Piza Arndt

Vitória, ES

2022

1. APRESENTAÇÃO DA PROPOSTA

O presente arquivo, conforme será tratado no próximo tópico, descreve o o objetivo do projeto e explica o seu código-fonte anexado conforme proposta abaixo.

Entre os temas dispostos, foi escolhido o tema VENDA. No código-fonte foram acrescentados comentários sobre a proposta implementada e referência ao respectivo tópico de acordo com as propostas a seguir:

- 1- Crie uma **classe** para representar o tema escolhido;
- 2- Crie uma **superclasse** para sua classe poder herdar;
- 3- Você pode criar outras **subclasses** de acordo com o objetivo do seu programa;
- 4- Crie **atributos genéricos** na superclasse;
- 5- Crie **atributos específicos** na classe de tema do trabalho;
- 6- Crie os **métodos set e get** para encapsular os atributos criados;
- 7- Crie métodos, e utilize em algum deles **estruturas condicionais ou de repetição**.
- 8- Utilize os seguintes comandos estudados na disciplina: **encapsulamento, herança, polimorfismo e sobrescrita**.
- 9- Destaque no seu código, **em comentários**, onde foi utilizado cada comando do item acima.
- 10- Descreva o objetivo do seu programa em um documento de texto, converta-o em PDF e encaminhe o seu código em um arquivo zipado, juntamente com o arquivo em PDF.

2. OBJETIVO

O objetivo do programa é mostrar ao final de cada compra uma mensagem de agradecimento, esta personalizada ao cliente que leve o seu nome no texto, a descrição do produto adquirido, assim como as informações de quantidade, preço, condição de pagamento e prazo para pagamento. No entanto, a depender da condição de pagamento, o valor final do produto pode ser alterado.

Para isso, primeiro foi criada uma classe abstrata, que não pode ser instanciada, apenas herdada, chamada “Movimento”, onde atributos genéricos foram criados, encapsulados e disponibilizados para acesso e manipulação por meio dos métodos getter e setter. A saber, os atributos dessa classe são: ‘tipo’ e ‘produto’ como ‘String’, ‘qtd’ como ‘int’ e ‘preco’ como ‘Double’.

Depois foram criados nesta classe dois métodos construtores, utilizando assim o conceito de polimorfismo por meio de sobrecarga (overload). O primeiro método precisa dos parâmetros ‘tipo’, ‘produto’, ‘qtd’ e ‘preco’. Todos serão atribuídos valores no momento da instanciação do(s) objeto(s) desta classe. Já o segundo método construtor, solicita apenas os parâmetros produto e quantidade quando da instanciação, sendo o ‘tipo’ previamente definido como “Bonificação” e o ‘preco’ igual a 0.0 (zero). Mais a frente, será explicado quando da necessidade de instanciar com cada um dos métodos.

Ainda na classe abstrata ‘Movimento’ foi criado o método ‘mostraInfo()’, que detalha as informações contidas nos 4 atributos desta classe e que será herdado por 3 subclasses, sendo duas delas sem que seja sobrescrito nas mesmas. Também o método ‘cobranca()’ foi criado para herança e sobrescrita na subclasse que for necessária.

Três subclasses foram criadas para herdar ‘Movimento’. São elas: ‘Venda’, ‘VendaPrazo’, ‘Bonificacao’.

Em ‘Venda’ foi criado o atributo específico chamado ‘cliente’ que receberá o texto com o nome do cliente. Herda o primeiro método construtor da classe abstrata e

adiciona como String o atributo cliente ao método construtor da subclasse, fazendo assim um override (sobreposição ou sobrescrita). Herda também o método 'cobranca()' sem sobrepô-lo. Foi criado nessa subclasse um método chamado 'mensagemVenda()' em 'Venda', assim como nas outras duas subclasses com uma mensagem personalizada de agradecimento com o nome do cliente.

Na subclasse 'VendaPrazo', também é herdado o primeiro método construtor e acrescentado também o atributo 'cliente' criado nesta subclasse. Além de 'cliente', o atributo 'prazo' foi criado e acrescentado como parâmetro no método construtor dessa subclasse. O método 'cobranca()' foi sobrescrito e criadas três variáveis e uma condicional 'if, else', para aumentar o preço final do produto de acordo com o prazo concedido ao cliente. Portanto, o cliente que compra com 90 dias ou mais de prazo, paga uma taxa de 8% a mais no produto. Quem compra com menos dias de prazo, paga apenas 5%. Lembrando que este método na subclasse 'Venda' não acrescenta nenhuma taxa de acréscimo ao valor final. Foi criado o método getter para o atributo prazo para que possa ser manipulado na instanciación.

A subclasse 'Bonificacao' herda o segundo método construtor da classe abstrata, que atribui valor zero a 'preco'. Sobrescreve o método 'cobranca()' retornando o próprio método da superclasse. Sobrescreve também o método 'mostraInfo()' alterando as informações do texto de impressão.

Por último, foi criada a classe principal chamada 'GestaoVenda', onde dentro do método principal de iniciação (main), foram instanciados um objeto para 'Venda', outros dois para 'VendaPrazo' e outro para 'Bonificacao'.

Segue o retorno ao compilar e rodar o programa:

Output - AtividadePOO (run)



```
run:
Caro(a) Caetano, agradecemos a preferencia!
Venda A Vista de 2 Notebook pelo preco de 3499.9 a unidade
O valor total a pagar e de 6999.8
-----
Caro(a) Maria Ortiz, agradecemos a preferencia!
Venda A Prazo de 1 Geladeira pelo preco de 1600.0 a unidade
O valor total a pagar e de 1680.0 com 60 dias
-----
Caro(a) Frida Calo, agradecemos a preferencia!
Venda A Prazo de 1 Geladeira pelo preco de 1600.0 a unidade
O valor total a pagar e de 1728.0 com 90 dias
-----
Caro(a) Domingos Gomes, agradecemos a preferencia!
Bonificacao de um TV 45'
O valor total a pagar e de 0.0
BUILD SUCCESSFUL (total time: 0 seconds)
```