



UNIVERSIDADE
FEDERAL DO CEARÁ

Relatório POO – AP3

Aluno : Pedro Anderson Costa Martins

Matrícula : 402454

Curso : Engenharia da computação

Avaliação Parcial 3

Nota = 10.0

Parte 1 (1 ponto) : Pontuação = 1 ponto.

Na primeira parte criei a superclasse abstrata Produto e as subclasses abstratas Bebida, PratoPrincipal, TiraGosto e Sobremesa. Na classe Produto fiz os atributos protegidos como pedidos na questão e criei o método abstrato obterValorUnitario e o método toString para retornar uma String organizada representando o produto.

Após isso, criei subclasses das subclasses abstratas, feitas anteriormente, para serem instanciadas. Os construtores são herdados das classes abstratas e recebem os atributos da classe como parâmetro.

Parte 2 (1 ponto) : Pontuação = 1 ponto.

Criei uma interface chamada Gerenciavel e fiz o método cancelarPedido. Coloquei a superclasse Produto para implementar a interface Gerenciavel.

Criei a classe ControleGerencial que possui o método cancelarPedido e por meio da interface Gerenciavel ele pode receber um pedido. Esse método altera o atributo cancelado do pedido para true.

Parte 3 (2 pontos) : Pontuação = 2 pontos.

Criei a classe itemPedido com os atributos produto e quantidade e um construtor que recebe os atributos como parâmetro. Também fiz o método obterValorItem para retornar a multiplicação do valor do produto do item pela quantidade consumida no produto no pedido.

Criei a classe Pedido com o atributo ArrayList de itemPedido e um boolean cancelado para ser verdadeiro caso o pedido seja cancelado. Também fiz os métodos adicionarPedido para adicionar pedidos a classe Pedido e o método obterValorTotalPedido para retornar a soma dos valores de todos os itemPedido. Para obter a soma utilizei o método obterValorItem.

Parte 4 (2 pontos) : Pontuação = 2 pontos.

Criei a classe BDProduto com um ArrayList de produtos disponíveis no restaurante, na mesma classe criei o métodos adicionarProduto para adicionar um Produto no array, retirarProduto para retirar um produto do array, procurarProduto para retornar um produto caso ele exista e null caso contrário e um toString para retornar uma String representando o produto. No método ProcurarProduto levantei a exceção ProdutoNaoEncontrado caso o produto requerido não esteja cadastrado no sistema.

Também criei a classe BDPedido com um ArrayList de pedidos realizados e criei os métodos adicionarPedido para adicionar pedidos no array, o método procurarPedido para retornar um pedido caso ele exista e null caso contrário e o toString para retornar uma String que representa o pedido.

Parte 5 (2 pontos) : Pontuação = 2 pontos.

Criei a classe de exceção ProdutoNaoEncontrado para ser levantado no método procurarProduto na classe BDProduto caso o produto não esteja na classe BDProduto.

Criei um método para enviar a mensagem “O produto de código X não está no sistema” quando o código do produto não estiver cadastrado no sistema.

Parte 6 (2 pontos) : Pontuação = 2 pontos.

Na main instanciei produtos de cada tipo disponível e os adicionei no BDProduto pelo método AdicionarProduto. Depois Listei todos os produtos disponíveis pelo método toString de BDProduto. Após isso fiz um do-while no qual o usuário insere o código do produto que ele deseja adicionar no carrinho e o programa confere pelo método ProcurarProduto no BDProduto se o código corresponde aos produtos cadastrados no sistema. Se conferir o programa pede ao usuário que insira a quantidade de itens do produto ele deseja colocar no pedido e pelo método adicionarItemPedido o programa insere o itemPedido no pedido e pergunta ao usuário se o mesmo deseja cadastrar um novo item, Caso contrário, ocorre tratamento de exceção visto que o produto não está cadastrado no sistema e a mensagem “O produto de código X não está no sistema” é impresso para identificar o processo. Após o pedido ser confirmado o valor total do pedido é impresso pelo método obterValorTotalPedido e em seguida o pedido é adicionado ao BDPedido pelo método adicionarPedido.

Fiz o teste de exceção durante o programa porém fiz outra vez após o término do mesmo para que o professor possa conferir. Inseri um valor que não está cadastrado no sistema para ocorrer o erro e imprimir a mensagem de erro.

Por último, busquei um pedido no BDPedido e instanciei a classe ControleGerencial e utilizei o método cancelarPedido com o pedido encontrado. Caso o pedido não esteja no sistema é impresso uma mensagem avisando que o mesmo não é existente.