

Alterar as classes ContaCorrente e ContaPoupanca
Acrescentar a elas a sobrescrita do método toString, que deve retornar uma String com todos os atributos concatenados. A classe ContaPoupanca deve reusar o toString da classe ContaCorrente.
Atributos – mediatorContaCorrente, do tipo MediatorContaCorrente
Classe TelaContaCorrente
Atributos – mediatorContaCorrente, do tipo MediatorContaCorrente
<p>Métodos:</p> <p>Construtor padrão.</p> <p><u>void iniciarTela()</u></p> <p>Ter como implementação a exibição de um menu com as opções de funcionalidades para conta corrente, a leitura no teclado da opção, as chamadas aos respectivos métodos auxiliares correspondentes a cada operação, e a saída do programa, caso seja digitada a opção 6.</p> <p><u>Métodos auxiliares privados:</u></p> <p><u>void incluir()</u></p> <p>Deve ler os dados de uma conta corrente do teclado, perguntar se a conta corrente a ser incluída é uma conta poupança, e se for, ler o atributo específico desta classe. Com os dados lidos, instanciar uma conta corrente ou uma conta poupança, chamar o método incluir do mediator, e verificar se a mensagem retornada é nula, indicando inclusão realizada com sucesso. Se for, mostrar a mensagem “sucesso na inclusão”. Se não for, mostrar a mensagem retornada.</p> <p><u>void creditar()</u></p> <p>Deve ler do teclado o número da conta e o valor a ser creditado. Com os dados lidos, chamar o método creditar do mediator, e verificar se a mensagem retornada é nula, indicando crédito realizado com sucesso. Se for, mostrar a mensagem “sucesso no creditar”. Se não for, mostrar a mensagem retornada.</p> <p><u>void debitar()</u></p> <p>Deve ler do teclado o número da conta e o valor a ser debitado. Com os dados lidos, chamar o método debitar do mediator, e verificar se a mensagem retornada é nula, indicando débito realizado com sucesso. Se for, mostrar a mensagem “sucesso no debitar”. Se não for, mostrar a mensagem retornada.</p> <p><u>ContaCorrente buscar()</u></p> <p>Deve ler do teclado o número da conta. Com o dado lido, chamar o método buscar do mediator, e verificar se a conta retornada é nula. Se for, mostrar a mensagem “conta não existente”. Se não for, mostrar os dados do objeto recebido através do comando System.out.println(conta), onde conta é a variável que recebe o retorno do método buscar, chamado do mediator.</p> <p><u>List<ContaCorrente> gerarRelatorioGeral()</u></p> <p>Chamar o método gerarRelatorioGeral do mediator, e imprimir em um loop os dados das contas correntes constantes na lista retornada, através do comando System.out.println(contas.get(i)), onde contas.get(i) é cada ocorrência da lista.</p>
Classe ProgramaTelaContaCorrente
Atributos: nenhum
<p>Métodos:</p> <p>main, já que a classe é um programa.</p> <p>Deve instanciar uma TelaContaCorrente e chamar, da tela instanciada, o método iniciarTela().</p>