

PRÁTICA SOBRE TRATAMENTO DE EXCEÇÃO

O código a ser manipulado neste exercício prático refere-se a um sistema básico utilizado por funcionários de uma agência bancária. Desse modo, a partir desse sistema o usuário pode criar contas e, a partir disso, sacar, depositar ou transferir dinheiro dessas contas.

1) Estude a implementação disponibilizada em **Codigo_v1**. Execute esse código e realize todas as operações possíveis disponibilizadas no menu de opções. A partir disso, responda as seguintes perguntas:

- a) O que acontece se o usuário escolher a opção “**3 – Depositar**” antes de criar qualquer conta? Por que?
- b) Considerando que o usuário não criou qualquer conta, quais outras opções do menu resultariam no mesmo problema ocorrido no item a)?

2) Estude a implementação disponibilizada em **Codigo_v2**. Execute esse código para responder as seguintes perguntas:

- a) Acontece algum erro quando o usuário escolhe a opção “**3 – Depositar**”, “**4 – Sacar**” ou “**5 – Transferir**” antes de criar qualquer conta? Por que?
- b) No caso de se escolher a opção “**3 – Depositar**” ou “**4 – Sacar**” ou “**5 – Transferir**” antes de se criar qualquer conta, como fica a solução apresentada do ponto de vista do usuário do sistema, ou seja, há como o usuário saber que a operação solicitada não foi realizada com sucesso?
- c) Corrija o problema detectado no **item b** fazendo com que os métodos da classe **Agencia** retornem valores (true/false) que indiquem para a classe cliente que a transação solicitada não pode ser atendida. Altere a classe cliente para que os retornos implementados nos métodos da classe **Agencia** sejam utilizados para informar o usuário o motivo pelo qual a transação solicitada não foi bem sucedida.

3) Estude a implementação disponibilizada em **Codigo_v3**. Complete os métodos **depositar**, **sacar** e **transferir** da classe **Agencia** para que os mesmos lancem exceções indicando o erro ocorrido. Faça o mesmo para o método **sacar** da classe **Conta**. Teste todas as funcionalidades do seu sistema para responder as seguintes perguntas:

- a) Ao tentar realizar um saque ou depósito em uma conta inexistente, qual mensagem o usuário recebe?
- b) Ao tentar realizar um saque em uma conta existente, mas cujo valor é maior do que o saldo da conta, qual mensagem o usuário recebe?
- c) Ao tentar realizar uma transferência a partir de uma conta inexistente, qual mensagem o usuário recebe?
- d) Ao tentar realizar uma transferência a partir de uma conta existente (que tenha saldo suficiente) para uma conta inexistente, qual mensagem o usuário recebe? Por que? Isso está correto? Nesse caso, o valor do saldo da conta de origem ficou errado após a tentativa de transferência? Por que?
- e) Para corrigir o problema levantado no **item d**, acrescente no bloco **catch** do método **transferir** da classe **Conta** o código `System.out.println(e.getMessage())` e execute novamente a sua aplicação tentando realizar uma transferência a partir de uma conta existente (que tenha saldo suficiente) para uma conta inexistente. Qual mensagem o usuário recebe agora? Por que?
- f) Para corrigir o problema levantado no **item e**, acrescente código no método **transferir** da classe **Agencia** para que uma exceção do tipo `RuntimeException` seja lançada quando a conta destino for inexistente. Teste novamente a sua aplicação para verificar se o problema foi corrigido.
- g) Agora tente realizar uma transferência a partir de uma conta existente (mas cujo saldo seja insuficiente) para outra conta também existente. Qual mensagem o usuário recebe? Isso está correto? Se não estiver correto, identifique no código onde está o problema e corrija-o.

4) Altere o código obtido após a resolução do **item g** da **questão 3** de modo que a exceção lançada para o caso de saldo insuficiente, ao se tentar sacar um valor de uma conta, seja realizada a partir da instanciação de um objeto de uma classe denominada **SaldoInsuficienteException** que você deverá implementar. Essa classe deve herdar da classe de exceções em Java denominada `Exception` e seu construtor terá como único parâmetro o número da conta, o qual deverá fazer parte da mensagem de erro.