

Práticas de Programação Orientada a Objetos

GAC106 – 2022-1

Exercícios de Conta Bancária - Parte 2

Versão: 2022-06-15



Nestes exercícios vamos continuar o Projeto de Conta Bancária que fizemos anteriormente.

O objetivo é praticarmos o uso de coleções.

Passo 1 – Coleção de Contas

No projeto de contas anterior, ao tratar a transferência entre as contas, nós criamos duas variáveis na classe principal para tratar duas contas diferentes. Mas agora que conhecemos o conceito de coleções, vamos substituir essas duas variáveis por uma lista de contas, utilizando a classe `ArrayList`.

Além disso, você deve adicionar uma opção no menu para listar as contas já criadas.

- Os dados de cada conta (número e nome do cliente) deverão ser exibidos em uma linha.

Lembre-se que no exercício anterior já alteramos nosso código para que o usuário informe o código da(s) conta(s) a ser(em) utilizada(s) nas operações de consultar saldo, saque, depósito e transferência.

- Agora você deverá alterar isso, para que a conta solicitada pelo usuário seja buscada na lista de contas.

Obs.: lembre-se que não devem existir duas contas com o mesmo número na lista de contas (o Passo 4 do exercício anterior garante que cada objeto conta tem um número diferente).

Passo 2 – Remoção de Contas

Vamos agora adicionar uma nova opção no menu do nosso programa para que o usuário consiga remover uma conta da lista.

- O usuário deverá informar o número da conta a ser removida.
- A conta só poderá ser removida se o seu saldo for igual a zero.
 - Se tiver dinheiro na conta, deve ser exibida uma mensagem para o usuário dizendo que não é possível cancelar contas com saldo disponível.
 - Se a conta estiver usando o limite (ou seja, tiver saldo negativo), deve ser exibida mensagem dizendo que não é possível cancelar contas em débito.

[Opcional] Passo 3 – Filtrar contas pelo nome do cliente

Deve ser adicionada uma nova opção de menu para filtrar as contas de acordo com o nome do cliente. Esta opção deve funcionar da seguinte forma:

- O usuário deverá informar uma string com o nome (ou parte do nome) de um cliente.
- O programa então exibirá os dados (número da conta e nome do cliente) das contas que tenham o nome (ou a parte do nome) informada pelo usuário.

Exemplo: Suponha que existam as seguintes contas (números de conta e nomes de clientes):

- 1000 - João da Silva
- 1001 - Joaquim Ferreira
- 1002 - Maria Pereira
- 1003 - Pedro José

E suponha que o usuário informe a string “Jo”.

- Nesse caso, deverão ser exibidos os dados das contas 1000, 1001 e 1003, pois em todas elas os nomes dos clientes possuem a substring “Jo”.

Utilize a **dica do próximo slide** para fazer este exercício.

Obs.: não é necessário tratar a questão de maiúsculas e minúsculas. Por exemplo, se o usuário digitar “jo” no exemplo acima, nenhuma conta será exibida.

Dica para o Passo 3

A classe String do Java possui um método chamado contains que indica se uma string possui uma determinada substring.

O programa abaixo ilustra a utilização do método contains.

- Experimente implementá-lo em seu computador.

```
Scanner entrada = new Scanner(System.in);

System.out.println("Digite uma frase:");
String frase = entrada.nextLine();

System.out.println("Digite uma palavra:");
String palavra = entrada.nextLine();

if (frase.contains(palavra)) {
    System.out.println("Palavra existe na frase");
}
else {
    System.out.println("Palavra não encontrada!");
}
```

[Opcional] Passo 4 – Entendendo as vantagens do HashMap

No Passo 1 nós criamos um ArrayList para gerenciar a coleção de contas no nosso programa. Agora veremos quais são as vantagens de se utilizar um HashMap para isso.

- Obs.: a implementação deste passo deve ser feita em outro projeto, pois ele vai alterar o que você fez nos passos anteriores e eu preciso ver os passos anteriores para corrigir seus exercícios. Portanto, você deve primeiro fazer uma cópia do seu projeto e chamar essa cópia de algo como *ContaBancariaHashMap*.

Neste segundo projeto você deverá:

- Substituir o ArrayList de contas por um HashMap de contas, no qual a chave é o número da conta e o valor é um objeto conta.
- Alterar a opção de listar contas para utilizar o HashMap.
- Alterar as opções de operação das contas (saldo, depósito, etc.) para buscar a conta do HashMap.
 - O que achou de como o código ficou em comparação com o anterior?
- Alterar a opção de remoção de contas para remover do HashMap.
 - O que achou de como o código ficou em comparação com o anterior?