Exercícios práticos

Nota: Adicionar o projecto no repositório por si criado em sua conta GitHub;
Utilizar boas práticas (documentação completa);
Utizar interface gráfica.

1. Crie uma classe abstrata ContaBancaria que contém como atributos o numero da conta, saldo e historico e como métodos abstratos sacar e depositar que recebem um parâmetro do tipo double.
2. Crie as classes Conta Corrente e Conta Poupança que herdam da Conta Bancaria. A primeira possui um atributo taxaDeOperação que é descontado sempre que um saque e um depósito são feitos.
   A segunda possui um atributo limite que dá credito a mais para o correntista caso ele precise sacar mais que o saldo. Neste caso, o saldo pode ficar negativo desde que não ultrapasse o limite. Contudo isso não pode acontecer na classe ContaCorrente.
3. Crie uma interface Imprimível que declara um método mostrarDados.
4. Faça as classes Conta Corrente e Conta Poupança implementarem a interface e na implementação do método mostre os atributos de cada conta.
5. Crie uma classe Relatório que possui um método gerarRelatório que receba um objeto imprimível e executa o método mostrarDados do objeto.
6. Crie uma classe executável na qual você instancia duas contas (uma de cada tipo), credita algum valor para elas e efetua um saque (obs: no objeto conta poupança, faça um saque maior que o saldo atual).
7. Crie um objeto relatório e execute o método gerar relatório para cada conta criada.
8. Incremente a classe ContaBancaria com o método transferir que recebe o parâmetro o valor (double) e um objeto conta bancaria e transfere o valor desejado da conta atual para cada conta informada. Use os métodos sacar e depositar para isso.
9. Crie uma classe Banco que possui um arraylist de contas bancárias e implemente os métodos inserir, remover e procurarConta.
   O primeiro e o segundo recebem um objeto conta (que pode ser corrente ou poupança) e o insere e remove no *ArrayList*, respectivamente. O terceiro recebe um inteiro como parâmetro representando o número da conta e retorna um objeto conta bancária, caso essa conta exista no *ArrayList*, ou null, caso contrário.
10. Faça a classe banco implementar a interface Imprimível, onde a implementação de método consiste em executar método mostrar dados de cada conta criada.
11. Crie outra classe executável para os testes que instancie um banco e ofereça o seguinte menu para o usuário:
    a) Criar conta: o usuário informa se é conta poupança ou corrente e os dados da conta. O objeto correspondente é criado e inserido no banco através do método inserir. Exibir uma mensagem de sucesso.
    b) Selecionar conta: o usuário informa o número da conta. Se a conta existir, mostra o menu abaixo. Caso contrário, mostra mensagem de conta inexistente.
    c) Remover conta: o usuário informa o número da conta ou seleciona. Se a conta existe, então ela é excluída e uma mensagem de sucesso é informada. Caso contrário, uma mensagem de conta inexistente é informada.

d) Gerar relatório: mostra os dados de todas as contas cadastradas no banco.
e) Finalizar: termina a aplicação. Se o usuário escolher a opção 2 mostre o seguinte menu:

a. Depositar: recebe um valor e deposita na conta.
b. Sacar: recebe um valor e tenta sacar da conta.
c. Transferir: recebe um valor e o número de outra conta. Caso a conta exista, transfere o valor de uma conta para a outra. Caso contrário, informar mensagem de conta inexistente.
d. Gerar relatório: mostra os dados da conta selecionada.
e. Retornar ao menu anterior: exibe o menu anterior (opções 1 a 5).