DESAFIO JAVA E ORIENTAÇÃO A OBJETOS

A seguir, você terá as informações para a realização do desafio de Java orientado a objetos. Atente-se ao passo a passo e aos requisitos para executar o desafio! Bons estudos!

CONTEXTO:

Você e sua equipe foram contratados pelo BANCO ITAÚ para desenvolver um sistema bancário. A equipe de negócios requisitou que o sistema deve possuir as seguintes funcionalidades:  
 1 – Cadastrar Conta

2 – Acessar Conta

3 – Sair

No cadastro de contas, cliente deve indicar se a conta que deseja abrir é do tipo Corrente ou Poupança. Após, deve ser solicitado ao cliente que informe:

1. Nome Completo: campo textual;
2. CPF : campo numérico;
3. Endereço: campo textual;
4. Telefone: campo textual;
5. Senha: campo numérico;
6. Saldo: campo numérico, com duas casas decimais.

Ao submeter estes dados o sistema deve gerar randomicamente:

1 – número de agência: campo numérico, com 4 caracteres

2 – número de conta: campo numérico, com 5 caracteres.

Requisitos Técnicos:

* Crie uma classe Pessoa (com atributos nome, cpf, endereço e telefone) e inclua o objeto como atributo na classe Conta.
* As classes de Conta Corrente e Conta Poupança são subclasses de Conta.
* Crie um objeto do tipo ContaCorrente ou ContaPoupanca conforme a indicação do cliente na criação de conta.
* A classe conta deve ser gravada em uma variável global e os atributos localmente.
* O sistema deve retornar o número da agência, conta e uma mensagem de sucesso.
* A aplicação não deve ser encerrada após criar a conta, deve retornar para o menu inicial.

No menu ACESSAR CONTA deve ser solicitado ao cliente que informe:

1 – agência;

2 – número da conta;

3 – senha

Caso o cliente digite agência, número da conta ou senha inválidos, deve-se retornar qual dado está incorreto. Após, valide se a conta do cliente existe.

Caso conta existente, exiba um menu com as seguintes funções:

DEPOSITAR

* Deve receber o valor do depósito, valide se é valor positivo, somar ao saldo e retornar o valor do saldo atualizado. Se for conta poupança, deve render 1% em cima do valor de depósito.

SACAR

* A aplicação deve receber o valor do saque, validar se o saldo for maior ou igual que o valor do saque solicitado. Valide também se é o valor positivo. Depois disso, subtraia o saldo pelo valor do saldo atualizado, senão apenas retorne mensagem de saldo insuficiente.

CONSULTAR SALDO

* Deve retornar o saldo da conta logada.

ENCERRAR A CONTA

* Deve retornar mensagem de confirmação.
* Se a resposta for sim, apaga a conta e volta para o menu inicial.
* Senão apenas retorne para o menu Acessar Conta.

Na opção SAIR do menu principal a execução da aplicação deve ser encerrada.

APROFUNDE-SE MAIS

- Dicas para pesquisar e incrementar a solução:

* Realize mais validações para garantir maior segurança no sistema.
* Crie outros tipos de contas e/ou novos métodos para outros tipos de transações.
* Faça tratamento de erros e exceções para deixar o código mais seguro.
* Implemente PIX como uma opção no menu, onde o valor da transferência é subtraído da conta logada e acrescido à conta de destino.